

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Detekce zvuku houkačky sanitního vozu v silničním
provozu**

Detection of Acoustic Emergency Signals in Road Traffic

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Josef Paleček**
Studijní program: N2649 Elektrotechnika
Studijní obor: 3901T009 Biomedicínské inženýrství
Téma: **Detekce zvuku houkačky sanitního vozu v silničním provozu**
Detection of Acoustic Emergency Signals in Road Traffic

Zásady pro vypracování:

1. Analýza stávajícího stavu problematiky a rešerše existujících řešení.
2. Návrh vlastního algoritmu pro automatickou detekci houkačky sanitního vozu ve standardním silničním provozu, jeho implementace a testování v prostředí Matlab.
3. Výběr vhodných senzorů a hardwarové platformy pro implementaci vytvořeného algoritmu a realizace funkčního prototypu pro testování navrženého algoritmu.
4. Testování navrženého algoritmu v reálných podmínkách silničního provozu.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] MCDONOUGH, Robert N. and Anthony D. WHALEN. *Detection of signals in noise*. 2nd ed. San Diego: Academic Press, c1995, xiv, 495 s. ISBN 01-274-4852-7.
[2] CHAPARRO, Luis F. *Signals and systems using MATLAB*. Burlington, MA: Academic Press, c2010, xvi, 752 p. ISBN 978-012-3747-167.
[3] HOROWITZ, Paul. *The art of electronics*. 2nd ed. New York: Cambridge University Press, 1989, xxiii, 1125 s. ISBN 05-213-7095-7.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Černý, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě, dne 10. 7. 2015

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke, positioned above a dotted line.

podpis

Na tomto místě bych rád poděkoval Ing. Martinu Černému, Ph.D., za odborné vedení, rady a připomínky, které mi v průběhu psaní této diplomové práce poskytnul. Také děkuji společnosti společnosti PRIME s.r.o. za poskytnutí zvukové nahrávky.

Abstrakt

Tato diplomová práce se zabývá problematikou detekce zvuku sanitní houkačky v silničním provozu. Práce je rozdělena do několika částí. Teoretická část se zabývá zpracováním signálů. Následně popsáno stávající řešení detekčních systémů sirén. Praktická část obsahuje analýzu reálného záznamu sanitní houkačky, a to v prostředí MATLAB pro offline záznam a v prostředí LabVIEW na platformě myRIO pro online záznam. Výstupem této práce je návrh softwarového algoritmu, který rozpozná sirénu v reálném prostředí.

Klíčová slova

Detekce sirény, FFT, zpracování signálu, MATLAB, Labview, myRIO

Abstract

This thesis deals with the detection of sound emergency siren in traffic. Thesis is divided to different parts. Theoretical parts deals with signal processing. Afterwards there is describe actual status of methods for detection emergency siren. Practical part of this thesis contains analysis of siren sound in computing program MATLAB. MATLAB is used for offline signal processing and myRIO platform is used for online signal processing of sound. Output of this thesis is software algorithm for detection emergency siren in real conditions.

Key Words

Emergency siren detection, FFT, signal processing, MATLAB, LabVIEW, myRIO

Seznam použitých symbolů zkratek

f_{vz} (Hz)	vzorkovací frekvence
f_{max} (Hz)	mezí (maximální) frekvence
f_{min} (Hz)	mezí (minimální) frekvence
$f_{tež}$ (Hz)	frekvence těžiště
$\Delta f_{tež}$ (Hz)	frekvenční rozdíl spektrálních těžišť
$\Delta f_{deltatež}$ (Hz)	frekvenční rozdíl $\Delta f_{tež}$ (Hz)
$R_{yx}(\tau)$	vzájemná korelace
τ	časový posun
l	délka segmentu
N	celkový počet vzorků
$s(t)$	spojitý signál
$T(s)$	základní perioda signálu
$t(s)$	čas
$w(n)$	váhovací okno
$\Delta t(s)$	časový rozdíl
$\Delta f(Hz)$	frekvenční rozdíl
DMA	(Direct Memory Access) přímý přístup do paměti
RT	(Real Time) zpracování reálného času
dB	jednotka Decibel
USB	Universal Serial Bus
FFT	(Fast Fourier Transform) rychlá Fourierová transformace
IIR	Infinite Impulse Response Filter
PC	(Personal Computer) osobní počítač

Obsah

1	Úvod.....	1
2	Popis a vlastnosti zvukového signálu.....	2
2.1	Vznik a šíření zvuku	2
2.2	Používané matematické metody	3
3	Stávající řešení detekčních systému.....	5
3.1	Příklady řešení detekčních metod	5
4	Návrh měřicího řetězce	8
4.1	Blokové schéma detekčního systému	8
4.2	Pořízení záznamu sirény	9
5	Analýza signálu v prostředí MATLAB.....	11
5.1	Analýza signálu v časové oblasti	11
5.2	Analýza signálu v kmitočtové oblasti	12
5.2.1	Segmentace záznamu sirény	12
5.2.2	FFT segmentů a spektrogram záznamu sirény	14
5.2.3	Filtrace signálu.....	18
5.2.4	Spektrální těžiště.....	21
5.2.5	Výpočet změny a průměrného zrychlení těžiště spektra	23
5.2.6	Vzájemná korelace spekter signálu	25
5.2.7	Střední krátkodobá energie segmentu sirény	28
5.3	Navržený algoritmus detekce sirény.....	28
6	Implementace detekčních metod v LabVIEW na platformě myRIO	32
6.1	Platforma NI myRIO	32
6.2	Přepočet audio raw hodnoty na napětí	33
6.3	Měřicí řetězec pro zpracování záznamu zvuku v reálném čase.....	34

6.4	Grafická uživatelská aplikace	35
6.4.1	Zobrazení zvukového záznamu v čase	36
6.4.2	Informační panel detekce sirény	36
6.4.3	Amplitudové spektrum signálu	36
6.4.4	Hrotová detekce amplitudového spektra	37
6.4.5	Průměrný počet detekcí sirény v časovém úseku	38
6.4.6	Výpis jednotlivých vypočtených parametrů	40
6.5	Navržený detekční algoritmus pro reálné zpracování signálu	40
7	Experiment	43
7.1	Zapojení testovacího řetězce	43
7.2	Průběh měření	44
7.3	Výsledky experimentu	45
8	Závěr	46
9	Literatura	48
10	Přílohy	49

1 Úvod

V současné době se automobilové koncerny snaží bojovat o zákazníky na mnoha polích, počínaje bezpečností vozidel vyrobených až po dodatečného vybavení a pohodlí. Automobily různých tříd cílí na různé koncové zákazníky. Například vozy vysoké třídy chtějí poskytnout mimořádné pohodlí během cestování. Obvykle je toho dosaženo použitím instalace klimatizace, pohodlnými sedadly, vysokou úrovní izolací hluku nebo moderními audio-systémy. Hlavní myšlenkou je izolovat cestující od vnějšího hluku, aby se necítili jako v křehké skleněné baňce. Pro cestující v autě to může být velice komfortní, ale nikdo se nezajímá o to, zda není ovlivněna interakce řidiče s okolím. Navíc výrobci automobilů rok co rok zlepšují komponenty použité v automobilech, snaží se, aby byly tišší a pohodlnější. Je reálné, že přijde doba, kdy řidič nebude schopen rozpoznat zvuky v jeho okolí, včetně sirény přicházejících záchranných složek.

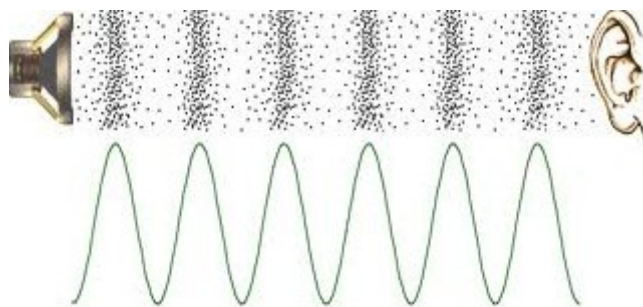
Složky záchranného systému mívají výrazné signální barvy, někdy s reflexními pruhy nebo battenburskou šachovnicí, světelné signalizační zařízení modré barvy (výstražný majáček) a sirény. Výstražná světla pracují skvěle spolu se zvukem zejména ve večerních, nebo nočních hodinách. V jasném denním světle mohou světla zůstat nezřetelná. Mohou zůstat neviditelná z důvodu nízkého kontrastu s okolím, nebo mohou být blokována jiným vozidlem. Hlavní metodou upozornění blížících se vozidel integrovaného záchranného systému je zvuk, který nevyžaduje, aby se sanitní vozidlo nacházelo v zorném poli řidiče.

A konečně je pro řidiče velice obtížné odhalit přítomnost vozidel integrovaného záchranného systému zvláště tehdy, kdy jsou všechna okna auta uzavřena a je puštěno autorádio. Tím je obtížnější pro řidiče reagovat včas a uvolnit prostor, a tím neomezit projíždějící sanitní vozidlo. Podobné situace nastávají každý den, a proto je důležité, aby se záchranné jednotky dostaly na místo nehody během několika minut, právě rychlost často může rozhodovat o životě a smrti postiženého.

2 Popis a vlastnosti zvukového signálu

2.1 Vznik a šíření zvuku

Zvuk obecně můžeme definovat jako mechanické kmitání, které je charakterizováno parametry pohybu částic pružného prostředí nebo u vlnového pohybu parametry zvukového pole. Část zvuků se projevuje jako slyšitelný zvuk což je akustické kmitání pružného prostředí v pásmu frekvencí od 16 Hz do 20 kHz, schopné vyvolat zvukový vjem. Frekvenční závislost definice slyšitelného zvuku je silně individuální, jen málokdo je schopen vnímat celé pásmo frekvencí (především horní hranice je velmi proměnná a závislá mj. na věku). Zvuky mimo toto pásmo neslyšíme, přesto jsme je schopni vnímat a mohou mít i nepříznivý vliv na zdraví či psychiku. Zvuky pod slyšitelnou hranicí (0,7 - 16 Hz) označujeme jako infrazvuk (velmi nízké frekvence, lidské tělo je vnímá hmatem jsou totiž schopny rozvibrovat celý povrch těla či bránici), zvuky nad slyšitelnou hranicí (do 50 kHz) jako ultrazvuk. [1]



Obrázek 1.: Zvuk a jeho šíření[1]

Ve volném prostoru se zvuk šíří od zdroje všemi směry volně a jeho šíření můžeme popsat tzv. vlnoplochami (spojnicemi všech míst zvukového pole, které mají v daný okamžik stejné parametry). Je-li zdroj zvuku malý (bodový), mají vlnoplochy tvar koule, je-li zdrojem např. rozměrná deska, jsou vlnoplochy rovinné (za rovinné považujeme i vlnoplochy ve velké vzdálenosti od bodového zdroje, kdy poloměr pomyslné koule už je tak velký že její výseč ve sledovaném bodě může být nahrazena rovinnou plochou). Šíření zvukových vln v prostoru popisuje celá řada fyzikálních principů. Vlny se v prostoru odrážejí, lámou i ohýbají, sčítají se s jinými vlnami, podléhají tlumení atd. [1]

2.2 Používané matematické metody

Diskrétní Fourierova transformace

Algoritmus byl vytvořen Cooleyem a Tukeyem a je nazýván „algoritmem decimování v čase“ nebo „algoritmem DIT“.

Používá se pro periodické signály diskrétní v čase. Délka záznamu signálu N lze rozložit takto: [2]

$$N = 2^m$$

Proto tuto posloupnost vzorků lze rozdělit na dvě části, a to se sudým a lichým pořadím.

Periodický signál diskrétní v čase má periodické spektrum (stejně jako kterýkoliv jiný signál diskrétní v čase). Vzorec pro přímou transformaci získá tvar [2]

$$\begin{aligned} F_k &= \sum_{i=0}^{N-1} x_i \exp(-j2\pi ki/N) = \\ &= \sum_{v=0}^{N/2-1} x_{2v} \exp(-j2\pi kv/(N/2)) + \exp(-j2\pi k/N) \sum_{v=0}^{N/2-1} x_{2v+1} \exp(-j2\pi kv/(N/2)) \end{aligned} \quad (1)$$

Vzájemná korelace

Jestliže je hodnocena závislost hodnot jednoho signálu ve dvou různých časových okamžicích, pak se jedná o autokorelační funkci.

V případě hodnocení závislosti dvou signálů ve dvou různých časových okamžicích, pak se jedná o křížovou (vzájemnou) korelační funkci.

Vzájemná korelační funkce signálů je závislá jen na rozdílu časových okamžiků t_1 a t_2 jako vzájemná korelační funkce stacionárních signálů. Ergodicita znamená možnost použití střední hodnoty součinu $x(t)y(t+\tau)$ v čase k výpočtu autokorelační funkce

$$R_{xy}(\tau) = \lim_{T \rightarrow +\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) y(t+\tau) dt \quad (2)$$

Vzájemná korelační funkce není sudá a její maximum nemusí být pro nulové posunutí $\tau = 0$.

Posunutí pro maximum křížové korelační funkce představuje dopravní zpoždění mezi signály. [2]

Spektrální těžiště

Spektrální těžiště je definováno jako těžiště velikostí spektrálních složek.

$$S_i = \frac{\sum_{k=0}^{K-1} M_i(j) \cdot j}{\sum_{k=0}^{K-1} M_i(j)} \quad (3)$$

Kde i je daný segment, j představuje spektrální složku, $M_i(j)$ je velikost spektrální složky a S_i reprezentuje danou frekvenci. [4]

3 Stávající řešení detekčních systému

V minulosti existovalo mnoho patentovaných principů pro detekci a včasné varování záchranných složek. Problematikou detekčních systémů se zabývá mnoho vědeckých a aplikačních skupin, avšak v současné době nebyly provedeny výzkumy v České republice žádné serióznější výzkumy. Všechny velké automotive společnosti se snaží orientovat své výrobky ke zvýšení bezpečnosti cestujících a zároveň jejich komfortu. Tento trend často předbíhá základní logické požadavky na bezpečnost silničního provozu. V posledních deseti letech vytvořily automobilové koncerny skupiny, které zaštiťují společnou cestu a standardy pro komunikaci mezi automobily a okolním prostředím silničního provozu.

Podle výzkumů z roku 1999 zabývající se zvukotěsnou izolací u moderních automobilů a útlumem sirény sanitního vozu. Sanitní vůz byl osazen sirénou, která produkovala intenzitu hluku o hladině 115dB. První test byl proveden u osobního automobilu v rychlosti 60 km/h, kdy autorádio a ventilace byly vypnuty. Siréna byla slyšitelná na vzdálenost 100 metrů od jedoucího auta. Se zapnutím ventilace a autorádia na minimum se slyšitelnost sirény rapidně snížila na 50 metrů od automobilu. Jakmile se hlasitost rádia zvýšila na hladinu 90dB, byl zvuk sirény o hodnotě 82dB téměř neslyšitelný. Zvuk houkačky sanitního vozu bylo možné slyšet až na vzdálenosti 2 metrů od automobilu.[8]

3.1 Příklady řešení detekčních metod

System to warn of an approaching emergency vehicles

Tento patent je založen na upozornění skrze radiový signál, který vysílá sanitní vůz. Standardní automobil pouze přijímá tento signál a upozorňuje na blížící se vozidlo.

Další metody pro zjišťování záchranných složek jsou založeny na měření zvukových nebo, optických signálů. Dle vybraného typu měření různé výzkumy rozšiřují využití na předávání varovné zprávy, detekci místa a vzdálenosti.

Integrated Circuit for Detection of Acoustic Emergency Signals in Road Traffic

Poukazuje na řešení detekce zvuku pomocí základních matematických operací, jako je konvoluce a FFT. Měření je založeno na zpracování a porovnávání nahraného signálu s předlohou. Následně dochází k aplikaci na křemík z důvodu velikosti, výkonu, proudového odběru a poměru SNR. Obvod ASIC vykazoval dobré vlastnosti, ale vývojová skupina upozorňovala, že je třeba dalšího testování v reálných podmínkách.[4]

Method for Automatic Driver Notification about Detection of Emergency Vehicles

Polská skupina vědců z Technické Univerzity v Lodži využila pro detekci signálů základních metod pro zpracování signálu. Byla použita metoda FFT a po její aplikaci byly vypočteny základní parametry spektra (maximální hodnota frekvence, minimální, průměr). Tento algoritmus byl poté aplikován na embedded platformu. Výsledky této metody zveřejněné v článku nebyly dostatečné, protože autoři nepočítali s Dopplerovým jevem a rovněž nebyla aplikována filtrace signálu. [5]

Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems

Jedná se o využití mikrofónového pole rozloženého do kříže. Znovu byly použity matematické metody, například korelace, metoda časového zpoždění zvuku, metoda nejmenších čtverců a adaptivního filtru, s následným porovnáváním metod a využitím pro detekci místa vzniku zvuku.[6]

C2C, EAR-IT

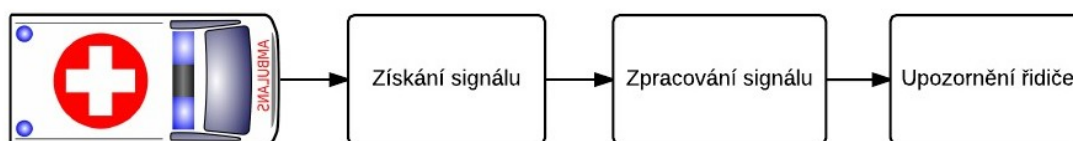
Stávající řešení v oblasti bezpečnosti automobilového provozu je směřováno skrze obecnou rovinu peer to peer komunikace mezi automobily. Organizace C2C (CAR 2 CAR Comucation Consortium) se například zabývá několika projekty, které řeší komunikaci mezi automobily a následné předání informace o dopravní situaci. Mají za úkol řešení standardů pro IT strukturu v automobilové dopravě atd.

Hlas budoucnosti ukazuje směrem k inteligentním městům a systémům včasného varování. Projekt EAR-IT se zaměřuje na propojení budov, dopravy a bezdrátové komunikace. Získané informace z dopravy jsou zpracovány a použity pro řízení semaforů a informování dopravních účastníků.

4 Návrh měřicího řetězce

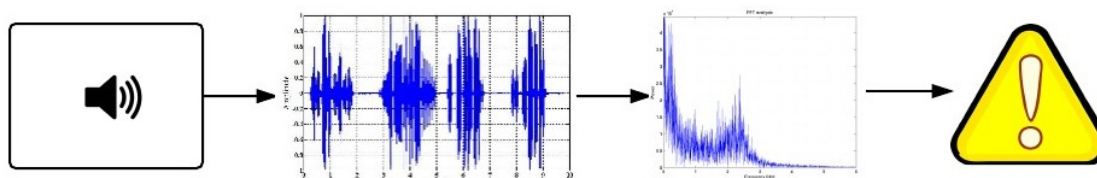
4.1 Blokové schéma detekčního systému

Pro realizaci systému na detekci sirény záchranných složek je nutné rozlišovat prostředí, ve kterém se řidič nachází. Je třeba si uvědomit, jak bezpečně řidiče upozornit na sanitní vůz v jeho okolí. Tedy vytvořit řidiči prostředí, v němž se může plně soustředit pouze na nadcházející situaci. Obecné blokové schéma této myšlenky je na zobrazeno obrázku č. 2.



Obrázek 2.: Obecný blokový diagram detekce vozů integrované záchranné složky

K tomu problému je možné přistupovat dvěma způsoby. Jedním z nich je detekování výstražných světel, druhým je detekce zvukové signalizace záchranné složky. Detekce zvuku je relativně jednodušší, protože k ní není potřeba vizuálního kontaktu. Zpracování zvuku má další nespornou výhodu v menší náročnosti na výpočetní výkon. Rychlost zpětné vazby na detekovaný v signál bude rychlejší než u obrazového zpracování.[5]



Obrázek 3.: Blokový diagram procesu detekce sirény integrované záchranné složky

Existuje více možností, jak řidiče upozornit na přibližující se vůz záchranné služby. Můžete přehrát zvuk sirény v kabině řidiče, rozsvítit informační ikonu na palubní desce či ztišit zapnuté autorádio. Blokové schéma procesu detekce sirény zachycuje obrázek č. 3.

Pro prvotní analýzu signálu a zpracování signálu byl použit program MATLAB. Metodou „rychlého vývoje“ byl aplikován algoritmus z prostředí MATLAB na platformu myRIO společnosti National Instruments.

4.2 Pořízení záznamu sirény

Je však na místě připomenout, že Česká republika nemá obecnou normu na tónovou křivku sirény u vozidel s právem přednostní jízdy. Právní předpis definuje pouze minimální hladinu hluku a skutečnost, že se musí jednat o spojitě proměnný tón. Sirény pohotovostních vozidel jsou postaveny na bílém šumu, který má schopnost pronikat prostředím a nezpůsobovat odraz zvuku. Tím je zamezeno vzniku ozvěny, která by ovlivňovala určení směru zdroje zvuku.

V České republice existuje mnoho výrobců, kteří mohou svůj specifický průběh tónové křivky sirény. Většina generátorů sirén obsahuje základní tóny.

WAIL

YELP

Hi-Lo

Prohouknutí

HORN pro zvýraznění výstrahy

Tyto generátory mohou automaticky po nastavení kombinovat módy: např. mód Hi-Low, kde siréna začíná v módu Hi-Low a automaticky se po 10 sekundách přepne do módu Yelp a naopak.

Je nutné si uvědomit, že pro základní rozbor a nalezení charakteristických vlastností sirény bylo třeba mít signál, který je ideálně čistý a bez rušení okolního prostředí. Takové rušení by se mohlo projevit ve zpracování a hledání charakteristických znaků signálu pro detekci.

Záznam byl pořízen díky společnosti PRIME, která nám ochotně nahrála minutový záznam typu WAIL. Siréna vzniká na generátoru firmy, kde vygenerovaný zvukový signál putuje do připojeného reproduktoru doporučeného výrobcem.[12]

Parametry generátoru:

Typ generátoru: IW1

Výkon: 100W

Provozní napájení: DC 12V

Způsob nahrání: Generátor – PC

Druhy tónových módu: WAIL, YELP a dále MANUAL a HORN

Mikrofon pro přenos hlasu: ANO

Možnost ovládání LED výstražných světel: ANO



Obrázek 4.: Obrázek generátoru sirény typu IWI[12]

5 Analýza signálu v prostředí MATLAB

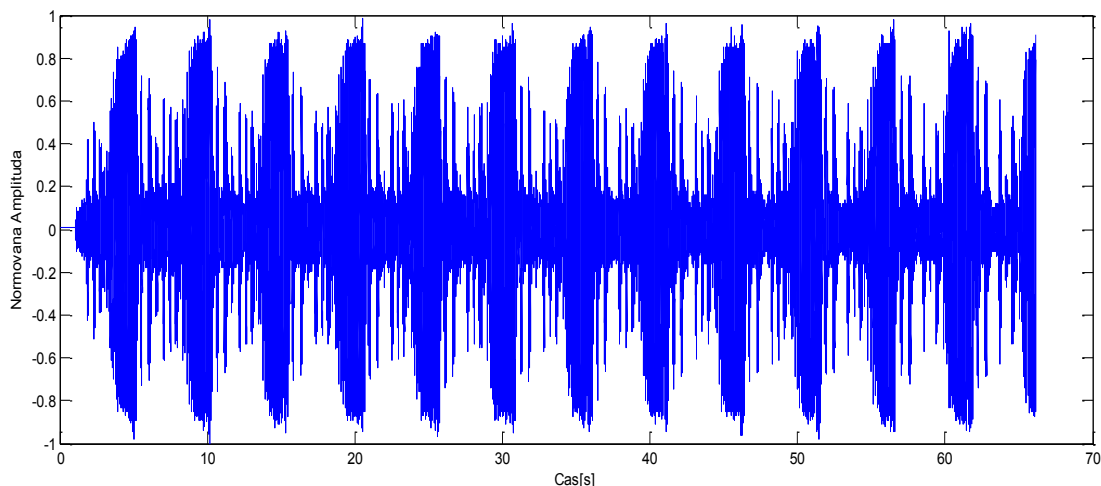
5.1 Analýza signálu v časové oblasti

MATLAB[®] je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. MATLAB je nástroj jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti. Výpočetní systém MATLAB se během uplynulých let stal celosvětovým standardem v oblasti technických výpočtů a simulací ve sféře vědy, výzkumu, průmyslu i v oblasti vzdělávání.[7]

Pro první základní zpracování signálu bylo třeba nahrát zvukový záznam do prostředí MATLAB. Pomocí příkazu *audioread* byl záznam nahrán a uložen do proměnné *signál*.

```
[signál,fs] = audioread('C:\...\MATLAB\us_wail60s.mp3');
```

Také byla získána vzorkovací frekvence o hodnotě $f_{vz} = 8000$ Hz. Signál je zaznamenán ve formátu .mp3 o délce 66 sekund. Jde o periodický signál s periodou $T = 5$ s. Signál je postupně kolísavý. Nahráný zvuk je vykreslen v obrázku č. 5.



Obrázek 5.: Záznam sirény typu WAIL

Z důvodu standardizace bylo před dalším zpracováním potřeba ustředit (odstranit DC složku signálu) a normalizovat vstupní data. Ustředěním signálu se

zbavíme stejnosměrné složky v signálu, která může ovlivnit hodnotu při výpočtu energie signálu. Zbavíme se jí pomocí odečtením střední hodnoty od signálu.

```
signal=signal-mean(signal); %ustredeni
```

Pomocí normalizace dat se udělí amplitudě signálu rozmezí 1 a -1. Podělením jednotlivých vzorků dat maximální absolutní hodnotou vzorku v datech dostaneme normalizovaný tvar signálu.

```
signal=signal/max(abs(signal)); %normalizace
```

5.2 Analýza signálu v kmitočtové oblasti

Pro počátečnou analýzu signálu jsme museli rozebrat signál ve frekvenční oblasti pomocí segmentace okenní funkci, FFT, spektrogramem. Po rozboru signálu v kmitočtové oblasti jsme se rozhodli využít jeho charakteristik pro detekci.

5.2.1 Segmentace záznamu sirény

Z hlediska stacionarity bylo třeba segmentovat signál, aby bylo možné rozlišovat rychlé změny signálu. Délka rámců by měla být dostatečně malá, aby bylo možné pokládat signál v daném úseku za stacionární, ale na druhé straně přijatelně velká, aby bylo možné dostatečně přesně odhadnout požadované parametry. Parametry rámců jsou délka, překrytí posun rámce. [9]

Rozhodli jsme se zvolit šířku $w(n)$ okna o $l = 128$ vzorků a to $s(t) = 16$ ms bez překrytí, kde fs je vzorkovací frekvence a $delkaseg$ je šířka okna $w(n)$.

```
delkaseg=round(fs*16*10^(-3));
```

Při segmentaci klade náš signál požadavek na dobrou časovou rozlišitelnost, ale dle Heisenbergova principu neurčitosti ztrácíme dobrou frekvenční rozlišitelnost. Signál je posazen na vysokých frekvencích, tak nás nemusí netrápit frekvenční krok 62,5 Hz. Jeden z dalších požadavků na šířku segmentu byla rychlost a nenáročnost zpracování počtu vzorků na embedded platformu. Ve stanovené šířce segmentu Pro segmentaci jsme použili Hammingovu okenní funkci pro své parametry. Často používané Obdélníkové okno, které má velice úzkou šířku přechodu, ale vzniká na jeho přechodě

k spektrálnímu prosakování (spectral leakage) ve větší míře než u ostatních oken, a proto není vhodné pro naše potřeby.

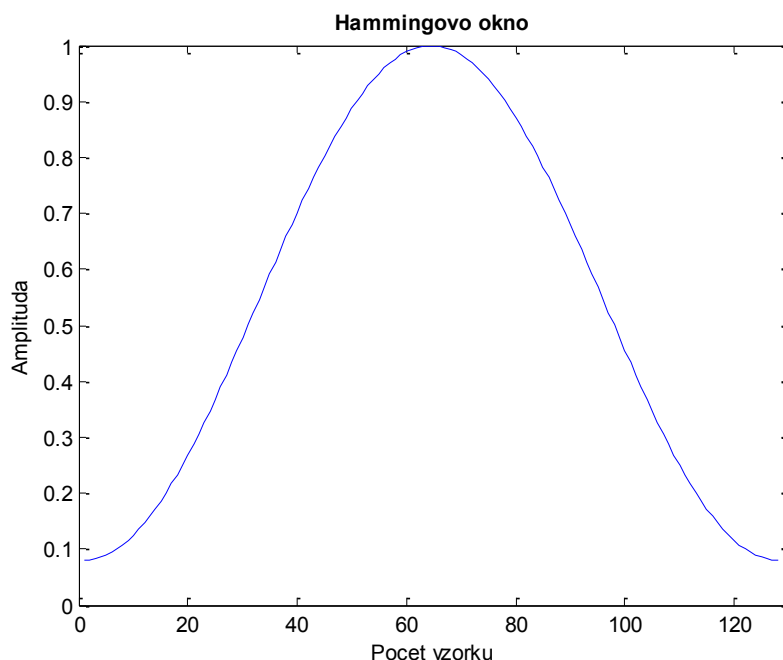
Vytvoření okenní funkce v MATLAB se vytváří pomocí příkazu *hamming*. *Delkaseg* je šířka okna $w(n) = 128$ vzorků.

`okno_hamm=hamming(delkaseg);`

Hammingovo okno $w(n)$ je definováno vztahem

$$w(n) = 0,54 - 0,46 \cos\left(2\pi \frac{n}{N}\right) \text{ pro } 0 \leq n \leq N - 1, \quad (4)$$

Kde n -tý vzorek diskrétního signálu je vyjádřen parametrem n . N je celkový počet vzorků diskrétního signálu.



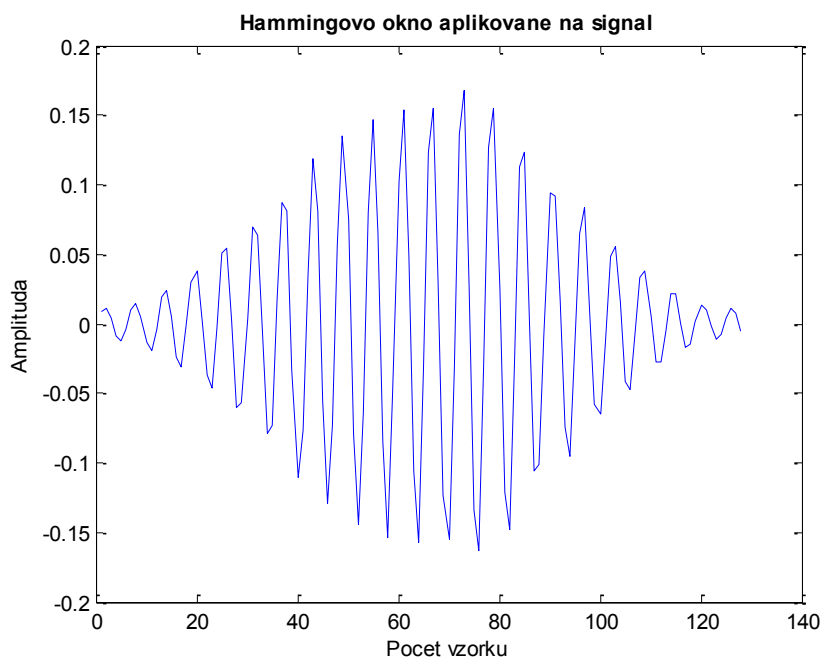
Obrázek 6.: Hammingovo okno o šířce 128 vzorků

Celý proces segmentace signálu na jednotlivé segmenty byl vykonán pomocí smyčky *for*. V této smyčce jsou vykonány další výpočty a úpravy signálu, které budou popsány v dalších kapitolách.

```
for i=1:pocetseg
DATA(i,:)=signal(((i-1)*delkaseg+1):i*delkaseg,1).*okno_hamm;%segmentace
DATA(i,:)=filter(hd,DATA(i,:)); %filtrace signalu
Energie(i,:)=(1/delkaseg)*sum(DATA(i,:).*DATA(i,:)); %stredni kratkodoba energie
end
```

Před celým procesem, bylo třeba vypočítat, na kolik segmentů, bude potřeba signál nasegmentovat. Toho jsme docílili podělením počtu vzorku signálu délkou segmentu. Pro ukázkou segmentu signálu po aplikaci Hammingova okna viz obr. č. 7.

```
[pocetvz,sloupcu]= size(signal);  
pocetseg=floor(pocetvz/delkaseg);
```



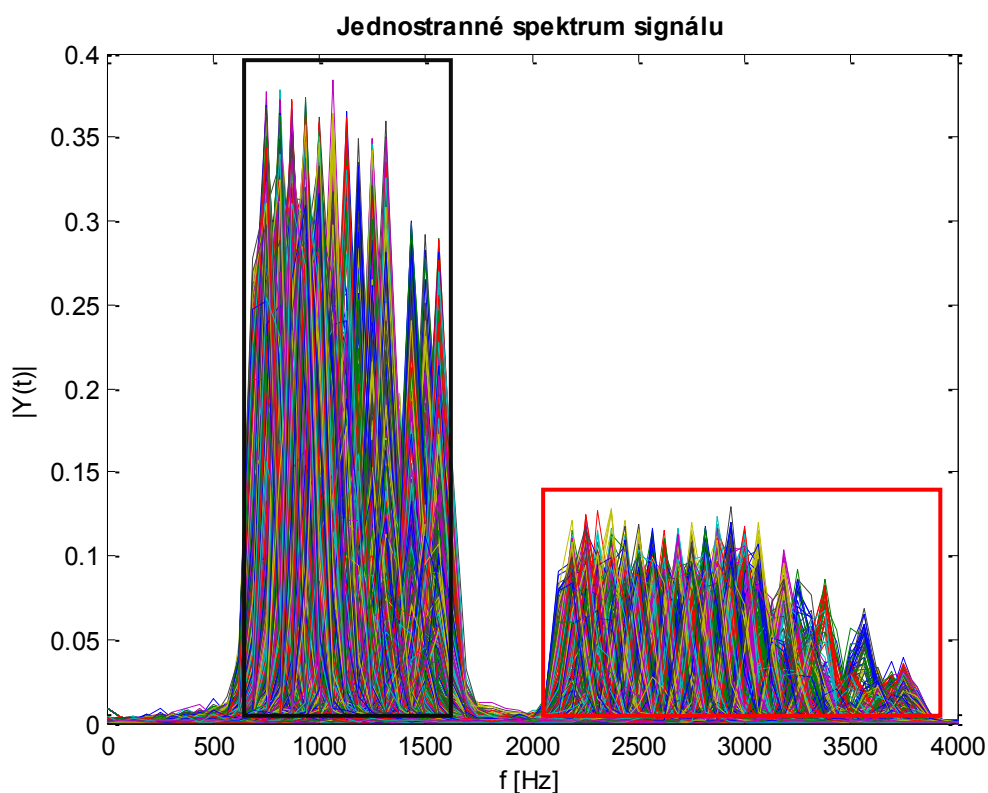
Obrázek 7.:Segment signál po aplikaci Hammingova okna

5.2.2 FFT segmenty a spektrogram záznamu sirény

Pro zobrazení spektrálních složek v jednotlivých segmentech jsme použili rychlou Fourierovu transformaci. Z FFT byla použita pouze jedna strana spektra, která má poloviční amplitudu. Pro správné zobrazení jsme celé spektrum vynásobili dvojkou.

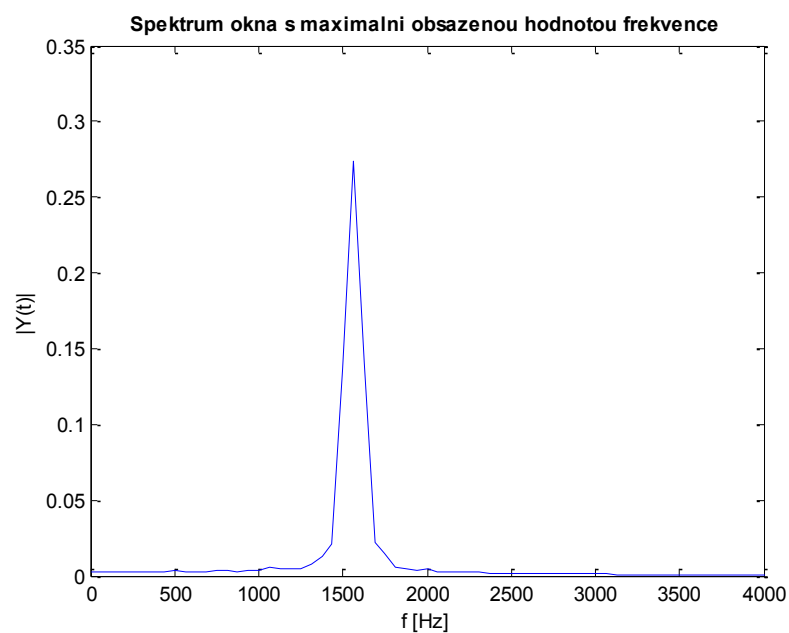
```
spektrum(i,:)=(2*((1/delkaseg)*(abs(fft(DATA(i,:))))));    % jednostranné  
spektrum
```

V obrázku č. 8 jednostranného amplitudového spektra lze vidět, že je signál v černém okně posazen na frekvencích v rozmezí $f_{sig} < 700; 1600 >$ Hz. V červeném okně můžeme pozorovat vyšší harmonické signálu. Tyto frekvence mohou být nežádoucí pro robustnost algoritmu na detekci signálu.

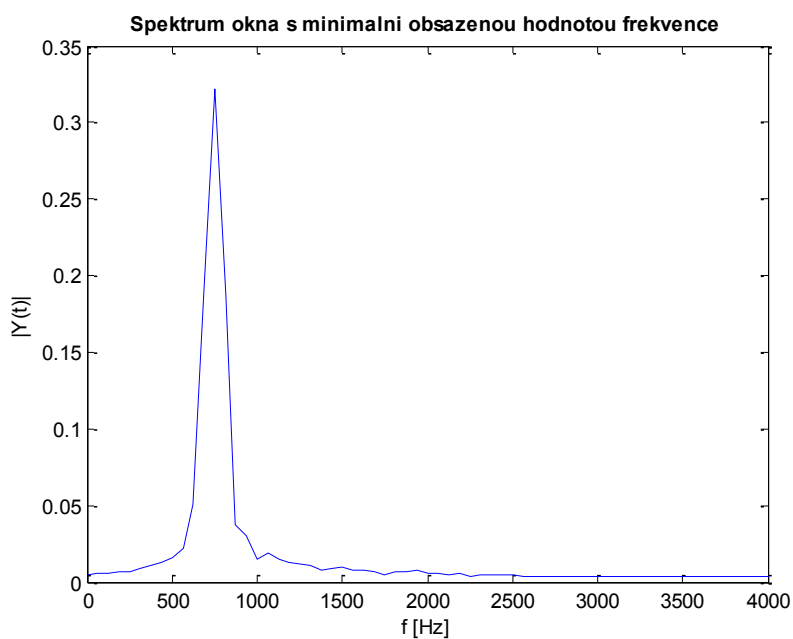


Obrázek 8.: Jednostranné amplitudové spektrum

Také bylo třeba vybrat okna $w(n)$ s maximální a minimální obsazenou hodnotou frekvence, které charakterizují frekvenční meze, v nichž se siréna pohybuje. Maximální vybraná mezní hodnota frekvence má hodnotu $f_{max} = 1563$ Hz a minimální mezní hodnotu frekvence $f_{min} = 750$ Hz. Tyto okna jsou zobrazena v obrázku č. 9 a 10.



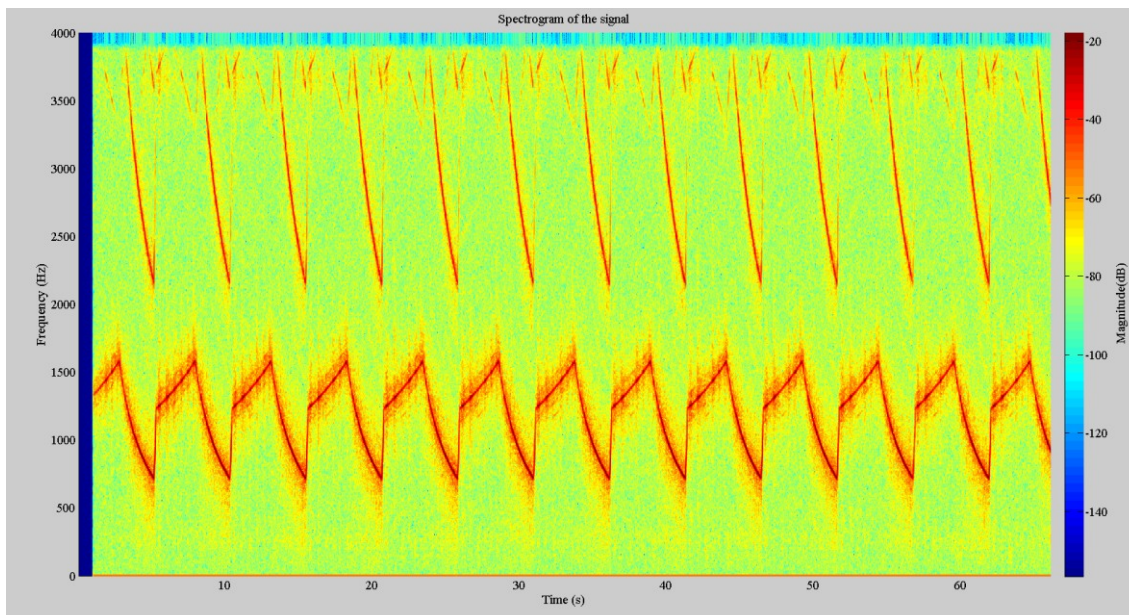
Obrázek 9.: Spektrální okno s maximální hodnou frekvence obsažené v siréně



Obrázek 10.: Spektrální okno s minimální hodnou frekvence obsažené v siréně

Díky vhodně zvolenému oknu můžeme vidět, jak dominantní spektrální komponenta ve fázi vzestupu frekvence sirény posunuje po ose x k vyšším frekvencím. Naopak během sestupné fázi se frekvence sirény dominantní spektrální komponenty

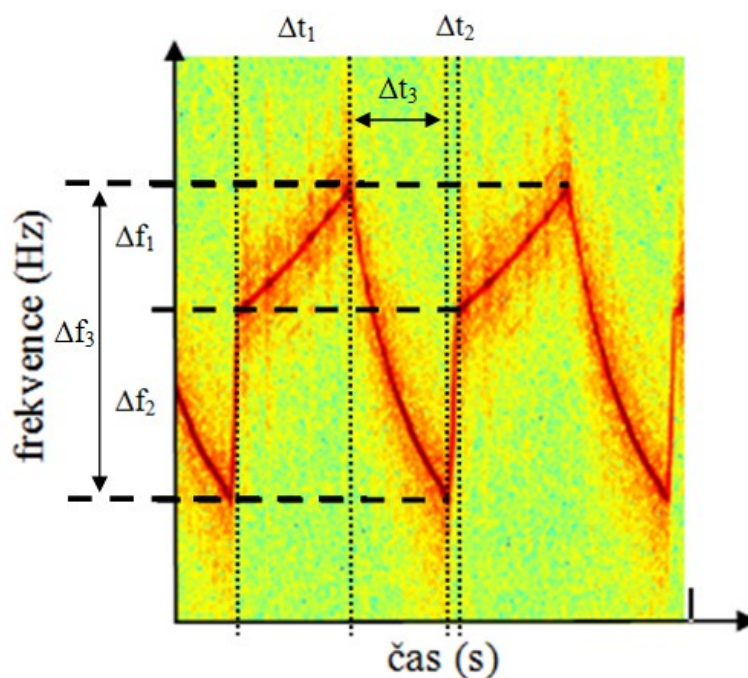
posunuje po ose x k nižším frekvencím. Nevadí nám ani frekvenční krok 62,5 Hz, protože frekvenční interval sirény je f_{sig} .



Obrázek 11.: Spektrogram záznamu sirény typu WAIL

Pro zobrazení frekvenčních složek signálů a jejich intenzity v závislosti na čase jsme použili spektrogram. Na obrázku č. 11 můžeme vidět, že se průběh lehce podobá pilovitému signálu. Velice čistě je zobrazená periodicitu signálu a také intenzita, která je uložena v rozsahu frekvenci f_{sig} .

Pro rozebrání bližších vlastností signálu je na obrázku č. 12 zachycen výřez části spektrogramu. Řádově v oblasti $\Delta f_1 = 370$ Hz a lze říct, že rychlost $\Delta t_1 = 2,7$ s změny frekvence signálu je v této oblasti velice lineární. Další charakteristická část signálu je $\Delta f_2 = 500$ Hz, kde skoková změna signálu je velice razantní a je také pro celý průběh signálu netypická a odpovídá skokové změně z času $\Delta t_2 = 0,010$ s. Hlavní sestupní část signálu, kde se tón sirény snižuje a trvá $\Delta t_3 = 2,5$ s, je o něco kratší než doba vzestupné části tónu sirény. Intenzita zvuku sirény se po celou dobu periody razantně nemění.



Obrázek 12.: Přiblížení časového úseku spektrogramu

5.2.3 Filtrace signálu

V kapitole 0 je zobrazen obrázek jednostranné rychlé Fourierovy transformace, kde lze vidět harmonické složky signálu sirény. Tyto komponenty signálu v budoucnu mohou dělat problémy při výpočtu spektrálního těžiště apod. Filtrací signálu zajistíme odstranění rušivých frekvencí a vytvoříme oblast zájmových frekvencí.

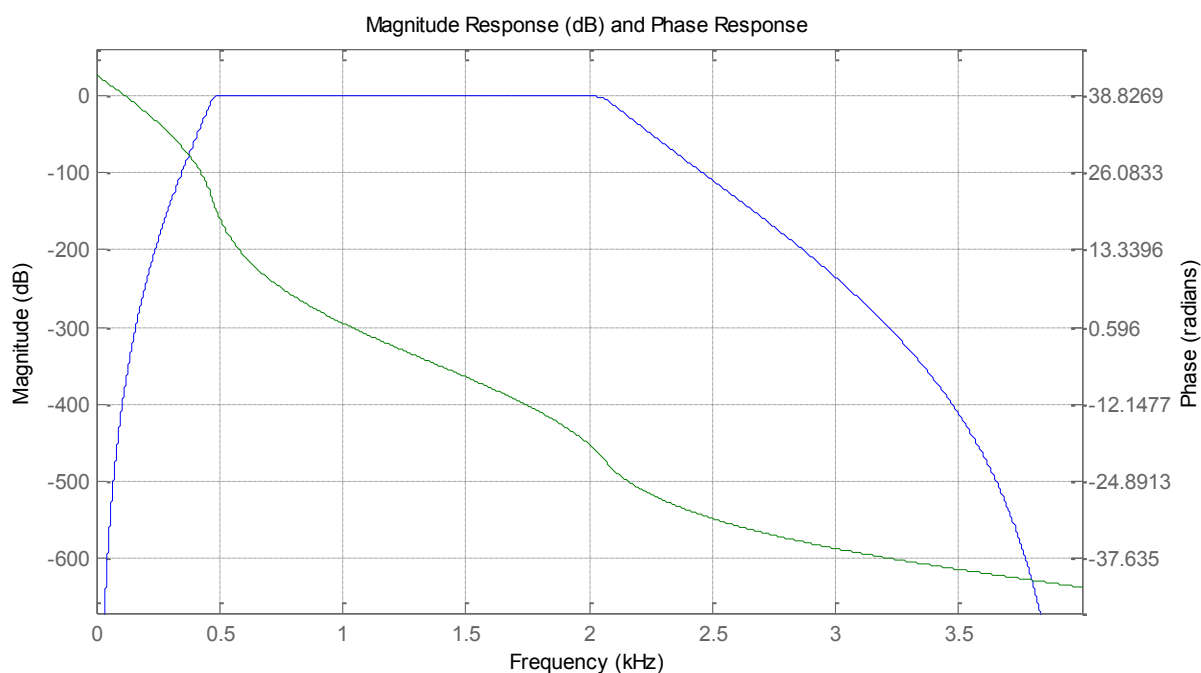
Pro návrh filtru byl použit toolbox programu MATLAB. Jde o Filter Design & Analysis Tool. Pro náš účel filtrace jsme použili digitální filtr IIR pásmovou propust' s aproximací Butterworth. Tento typ filtru jsme vybrali pro jeho plochou amplitudovou charakteristikou. Parametry filtru jsou uvedeny v tabulce č. 1.

Tab 1.: Parametry filtru

Typ filtru	Pásmová propust'
Použita aproximace	Butterworth
Řád filtru	54
Propustné pásmo (Hz)	500 2000
Zádržné pásmo (Hz)	350 2300

Zvlnění v propustém pásmu (dB)	0,1
Potlačení pásma (dB)	60
Útlum o - 3 dB první hrana (Hz)	515
Útlum o - 3 dB druhá hrana (Hz)	2060

Frekvenční a fázová charakteristika navrženého filtru je zobrazena na obrázku č. 13.



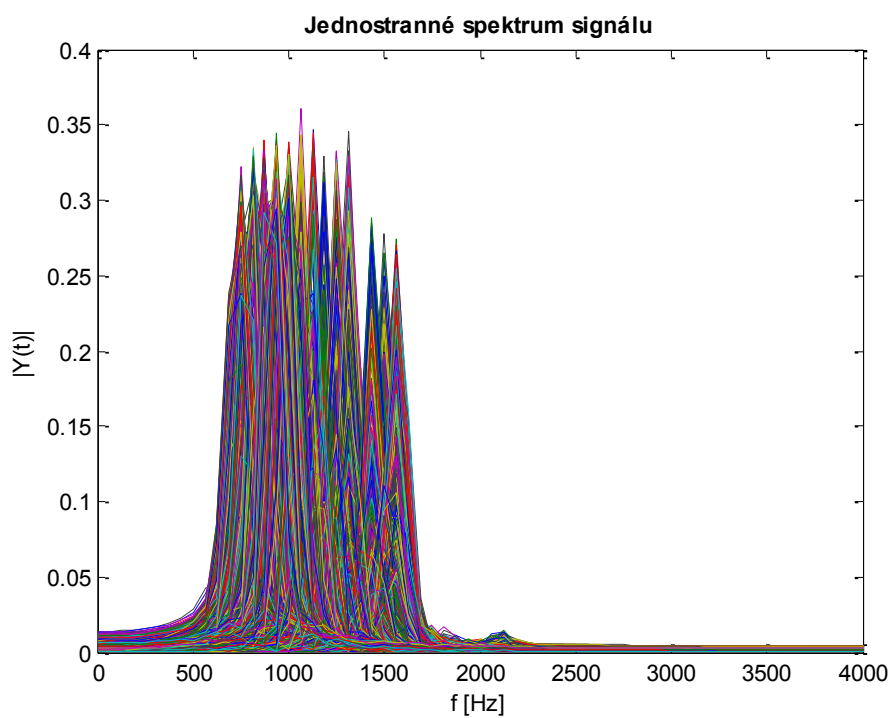
Obrázek 13.: Frekvenční a fázová charakteristika filtru typu Butterworth
pásmová propust'

V našem případě jsme potřebovali přenést konečný návrh filtru z toolboxu. Využili jsme funkci `fdesign.bandpass`, která vytvořila filtr s požadovanými parametry.

```
Fp = [500 2000]; % propustne pasmo
Fs = [350 2300]; % frekvence utlumeni
Fvz = fs; % vzorkovaci frekvence
Ap = 0.1; % zvlneni v propustnem pasmu [dB]
As = 60; % potlaceni mezi Fp a Fs [dB]
d=fdesign.bandpass(Fs(1),Fp(1),Fp(2),Fs(2),As,Ap,As,Fvz);
hd=design(d,'butter'); %navrh filtru o danyh vlastnostech
```

Navržený filtr jsme aplikovali ve smyčce *for*, kde probíhá segmentace signálu.
`DATA(i,:)=filter(hd,DATA(i,:));`

Na každý segment signálu byl aplikován filtr pásmová propust'. S aplikací filtru byly odstraněny vyšší harmonické a ostatní nežádoucí kmitočty viz obrázek č. 14.



Obrázek 14.: Jednostranné amplitudové spektrum po aplikaci filtru

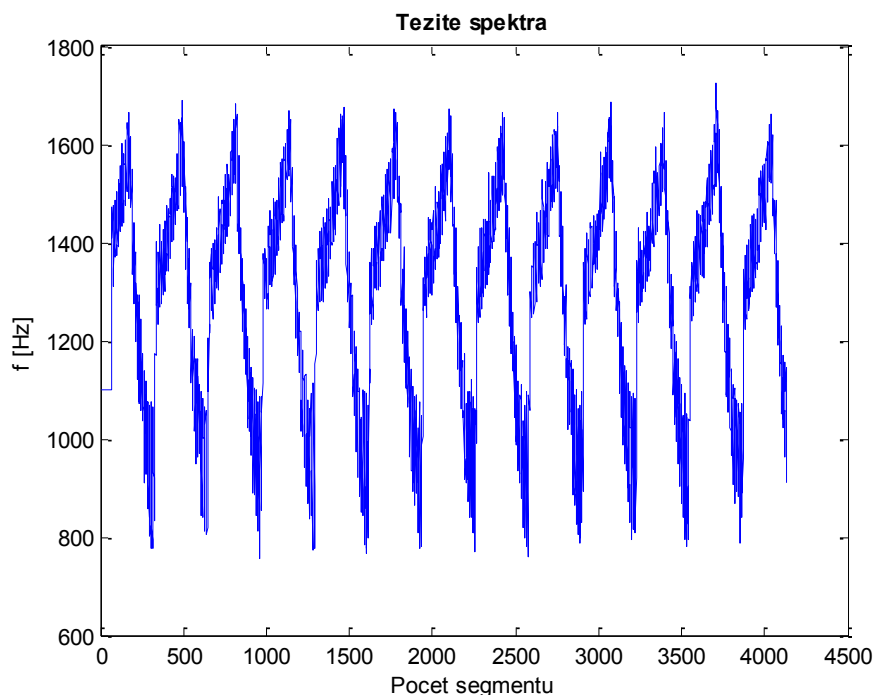
5.2.4 Spektrální těžiště

Ze spektrální analýzy a charakteristiky signálu tušíme, že v jednotlivých segmentech můžeme očekávat dominantní amplitudu, která bude charakteristická pro zvuk sirény. Podle této myšlenky jsme na signál aplikovali výpočet spektrálního těžiště. Spektrální těžiště se vypočítá jako vážený průměr frekvencí, které byly stanoveny pomocí FFT a amplitudami jako jeho váhami. Charakteristický hrot sirény tedy bude mít největší váhu ve výpočtu a těžiště by se nemělo rozhodit nežádoucími frekvencemi, které mají menší váhu.

```
teziste(i,1)=(sum((spektrum(i,1:delkaseg/2)).*(0:((delkaseg/2)-  
1))))/(sum(spektrum(i,1:delkaseg/2))); %vazeny prumer spektralni teziste
```

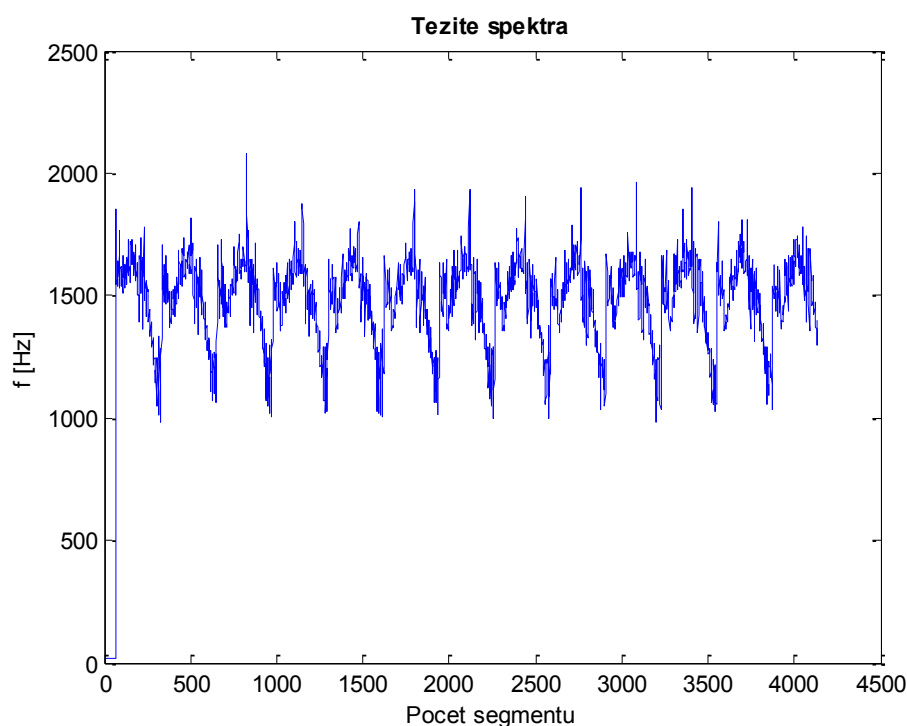
```
t_f(i,1)=teziste(i,1).*(fs/delkaseg); %prepocet na frekvenci
```

Přiložená část kódu počítá spektrální těžiště jednotlivých oken a přepočítává vzorky na frekvenci a ukládá hodnoty do proměnné t_f .



Obrázek 15.: Těžiště spektra záznamu sirény

Obrázek č. 15 zobrazuje jednotlivé vypočtené těžiště spekter celého signálu sirény. Vidíme, že těžiště kopíruje průběh spektrogramu sirény a frekvence těžišť se pohybuje v rozmezí $f_{t\check{e}z} <800;1650>\text{Hz}$. Při vypnutí filtrace segmentu, lze na obrázku č. 16 vidět změnu v těžišti spekter. Změna byla způsobena vyššími harmonickými signálu, které jednotlivá těžiště frekvenčně posunuly. Z tohoto důvodu bylo nutné filtrovat jednotlivá okna segmentace.



Obrázek 16.: Těžiště spektra záznamu sirény bez aplikace filtru

V obrázku č. 16 lze stále rozeznat periodicitu sirény, ale průběh je deformován oproti filtrovanému spektrálnímu těžišti deformován. Frekvence je lehce posunutá a pohybuje se v rozmezí $f_{t\check{e}z} <1000; 1800>\text{Hz}$.

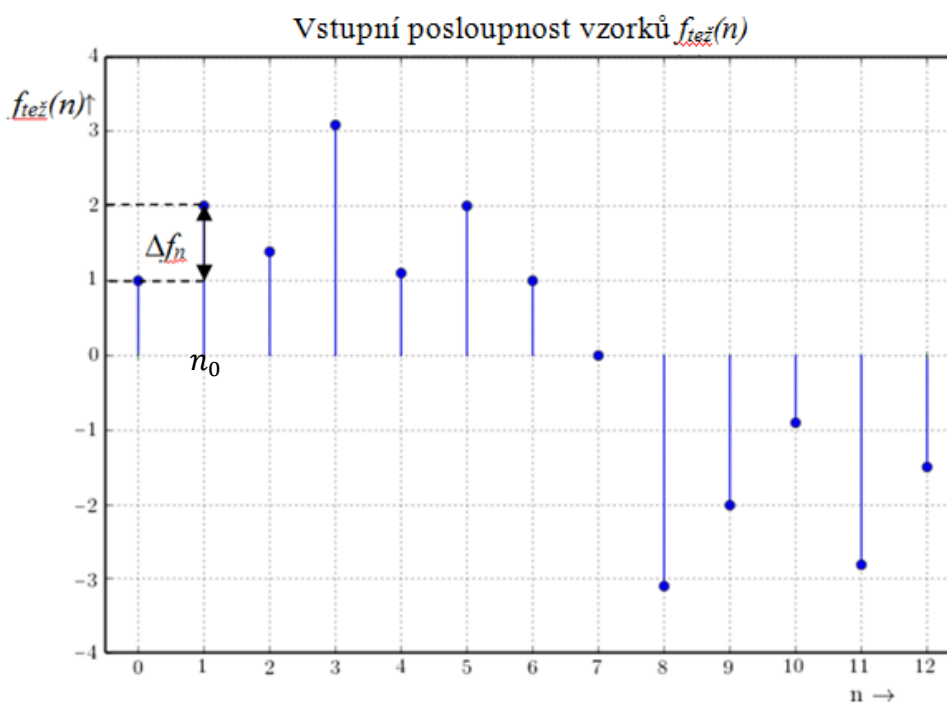
5.2.5 Výpočet změny a průměrného zrychlení těžiště spektra

Podle charakteru signálu a jeho lineárního vzestupu či sestupu frekvence, bylo vhodné analyzovat rychlost růstu, nebo pokles spektrálního těžiště. Pomocí tohoto rozdílu ve frekvenčních krocích můžeme odstranit signály, které mají rychle měnící se charakter a vytvořit hranice pro detekci sirény.

$$\Delta f_{tež}(n+1) = f(n+1) - f(n) \quad (5)$$

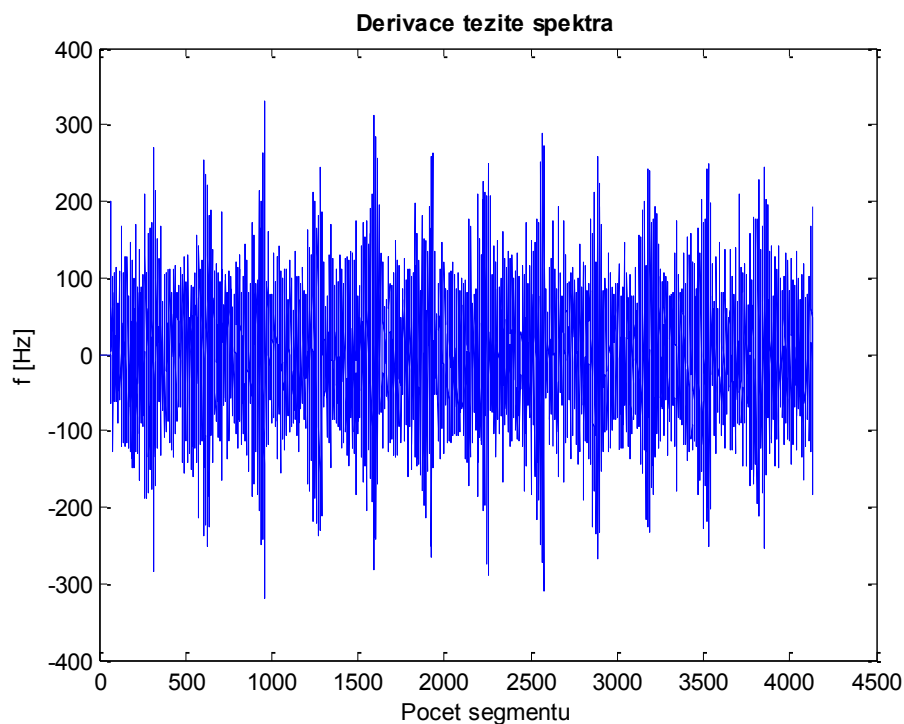
$$n = (n_0 + 1) - n_0$$

Na obrázku č. 17 můžeme vidět ukázkou posloupnosti vzorků a vypočtený rozdíl mezi nultým a prvním vzorkem.



Obrázek 17.: Grafický postup výpočtu změny spektrálního těžiště

Na obrázku č. 18 můžeme sledovat, v němž se velikost změny frekvence těžiště sirény pohybuje. Rychlost změn frekvence těžiště se pohybuje v rozmezí -150 až 150 Hz. Výrazné vrcholky vypovídají o velké změně frekvence těžiště spektra. Tyto změny odpovídají začátku nové periody signálu, kde je skokově velká změna frekvence.



Obrázek 18.: Výpočet změny těžiště spektra

Z výpočtu rozdílu dvou spektrálních těžišť je, také možné zjistit relativní zrychlení těžišť. Toto zrychlení nám může pomoci odstranit rychle se měnící signály a omezit chybnou detekci sirény. Výpočet vychází ze vztahu:

$$\Delta f_{\text{deltatež}}(n+1) = [f(n+1) - f(n)] - [f(n) - f(n-1)] \quad (6)$$

Výsledná úprava vzorce:

$$\Delta f_{\text{deltatež}}(n+1) = f(n+1) - 2 * f(n) + f(n-1) \quad (7)$$

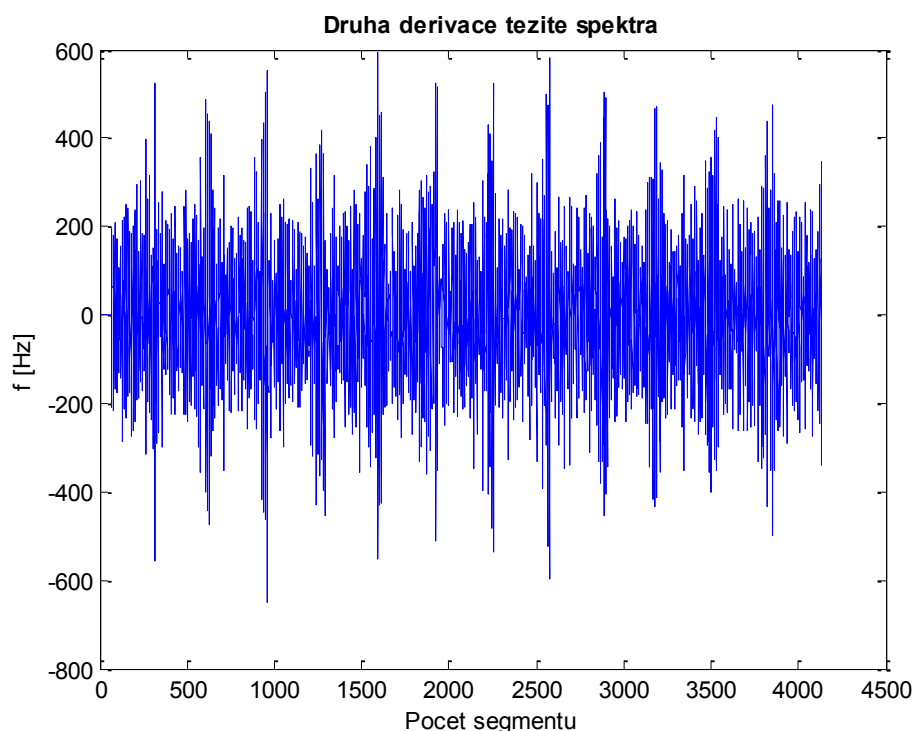
Následné vzorce byly převedeny do MATLAB skriptu pro zpracování signálu. Uložené výsledky byly uloženy do proměnných typu pole `delta_t_f` a `zrychleniDelta_t_f`.

```

if i >= 2 && i <= pocetseg
    delta_t_f(i,1)=(t_f(i,1) - t_f(i-1,1)); %rychlost zmeny frekvence teziste spektra
if i >= 3 && i <= pocetseg
    zrychleniDelta_t_f(i,1) = (t_f(i,1) - 2*(t_f(i-1,1)) + t_f(i-2,1)); %zrychleni teziste
spektra
end
end

```


Na obrázku č. 19 lze vidět, že průměrné zrychlení těžiště signálu se pohybuje v rozmezí -200 až 200 Hz. Znovu se objevují vrcholky, které vykazují velkou a náhlou změnu frekvence těžiště. Odpovídají začátkům nové periody sirény.



Obrázek 19.: Průměrné zrychlení těžiště spektra

5.2.6 Vzájemná korelace spekter signálu

Vzájemná korelace je v praxi často používána pro porovnání dvou signálů a určení jejich podobnosti. V našem případě použijeme vzájemnou korelaci pro zjištění pohybu nejvyššího vrcholu spektra po frekvenční ose.

Pro získání korelace bylo třeba segmenty jednotlivých spekter vzájemně „korelovat“ porovnat se spektry nejvyšší a nejnižší obsaženou hodnotou frekvence signálu viz

FFT segmenty a spektrogram záznamu sirény. Příkazem *xcorr* v prostředí MATLAB byla provedena vzájemná korelace spektrálního segmentu a segmentu s nejvyšší (nejnižší) frekvencí obsaženou v záznamu sirény.

%corelace jednostranneho spektra nahraneho signalu a minimalnich,hodnot spekter

```

for i=1:pocetseg
[amplitudyMin(:,i),tauMin(:,i)]=
xcorr(spektrum(i,1:delkaseg/2),spektrummin2(1,1:delkaseg/2));
[amplitudyMax(:,i),tauMax(:,i)]=
xcorr(spektrum(i,1:delkaseg/2),spektrummax2(1,1:delkaseg/2));
end

```

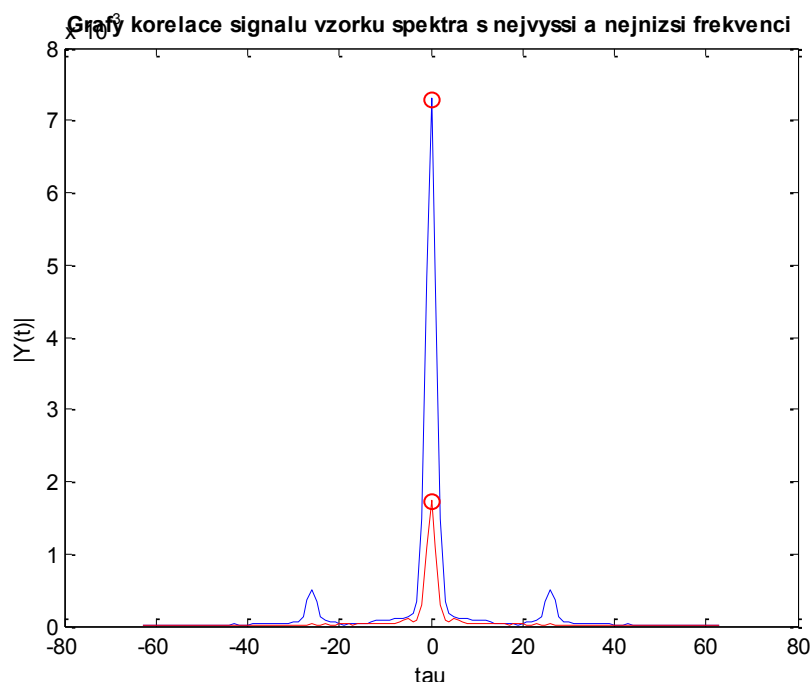
Korelace všech spektrálních segmentů byla provedena jak pro spektrum s nejnižší, tak s nejvyšší obsaženou hodnotou frekvence signálu. Po provedení křížové korelace jednotlivých spektrálních segmentů jsme výsledky uložili do proměnných typů pole. Jde o *amplitudyMin* a *amplitudyMax*, kde odpovídající posuny τ jsou uloženy v *tauMin*, *tauMax*. Pro získání největší amplitudy v korelovaném segmentu bylo nutné vyhledat tyto maxima pomocí funkce *max*. Vyhledané maximum korelovaného segmentu je uloženo do proměnné *maxAmpcorr* a odpovídající vzorek okna je uložen do *vzorekAmpMax*. Hodnota v proměnné *vzorekAmpMax* je použita pro extrahování hodnoty τ posunu korelovaného segmentu. Hodnota posunu τ je uložena do proměnné *corrTauMax*. Stejný postup je aplikován na všechny segmenty záznamu signálu.

```

for i=1:pocetseg
[maxAmpcorr,vzorekAmpMin]=max(amplitudyMin(:,i));
corrTauMin(i,1)=tauMin(vzorekAmpMin,i); %hodnota tau vzorku
[maxAmpcorr,vzorekAmpMax]=max(amplitudyMax(:,i));
corrTauMax(i,1)=tauMax(vzorekAmpMax,i);
end

```

Na obrázku č. 20 můžeme vidět výsledky vzájemných korelací dvou stejných spektrálních segmentů. Protože jde o korelace dvou stejných segmentů, největší známka překrytí spekter je na pozici s nulovým posunem τ .



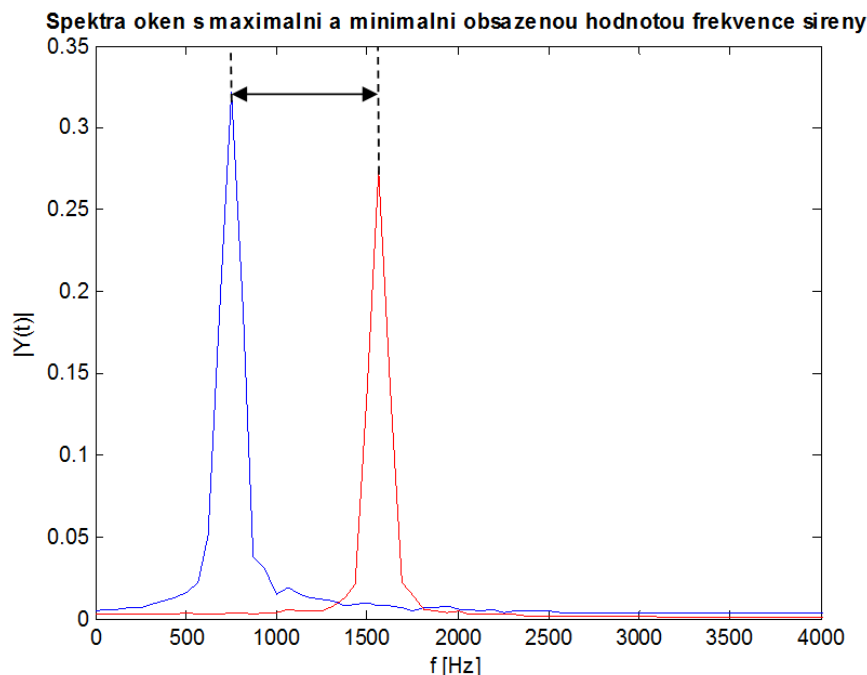
Obrázek 20.: Výsledek vzájemné korelace dvou stejných spektrálních segmentů

Přepočet hodnot posunu τ na frekvence se děje pomocí vzorce viz. níže.

$$\begin{aligned} \text{corrTauFreq} &= \tau * \frac{f_s}{l} \\ \text{corrTauFreq} &= 0 * \frac{8000}{128} = 0 \text{ Hz} \end{aligned} \quad (7)$$

`corrTauMaxFreq=corrTauMax.*(fs/delkaseg);`
`corrTauMinFreq=corrTauMin.*(fs/delkaseg);`

Proměnné `corrTauMaxFreq` a `corrTauMinFreq` obsahují přepočty hodnot `corrTauMax`, `corrTauMin`. Proměnné `corrTauMaxFreq` a `corrTauMinFreq` obsahují hodnoty posunu nejvyššího hrotu spektra oproti segmentům obsahující nejvyšší (nejnižší) hodnotu frekvence sirény. Výsledné hodnoty byly použity pro určení rozsahu posunu τ , ve kterém se spektrální segmenty sirény pohybují. Na obrázku č. 21 jsou zobrazena spektra segmentu, která obsahují nejvyšší a nejnižší frekvenci obsaženou v siréně. Pomocí vypočtených korelačních mezí jsme schopni říct, zda se spektrum jednotlivých segmentů pohybuje v oblasti frekvencí sirény, a pokud ne, nepůjde o sirénu.



Obrázek 21.: Hranice spekter segmentu, obsahující nejvyšší a nejnižší frekvenci obsaženou v siréně

5.2.7 Střední krátkodobá energie segmentu sirény

Výpočtem tohoto parametru jsme zjistili, v jakém rozmezí se pohybuje energie sirény jednotlivých segmentů. Rozmezí krátkodobé energie můžeme využít pro oddělení segmentů ticha a sirény. Energie jednotlivých segmentů je uložena v proměnné *Energie*. [10]

Energie sirény jednotlivých segmentů se pohybuje v rozmezí 0,0001 a 0,1. Výpočet energie sirény byl v prostředí MATLAB byl aplikován takto:

`Energie(i,:)=(1/delkaseg)*sum(DATA(i,:).*DATA(i,:)); %stredni kratkodoba energie`

5.3 Navržený algoritmus detekce sirény

Z předcházejících výpočtů, těžiště spektra, korelace, zrychlení těžiště spektra atd. byly vytvořeny hranice, které jsou signifikantní pro detekci sirény. V tabulce č. 2 můžeme vidět jednotlivé hranice vypočtených parametrů signálu sirény. Pro splnění podmínky detekce musí každý jednotlivý segment splňovat meze, které vycházejí

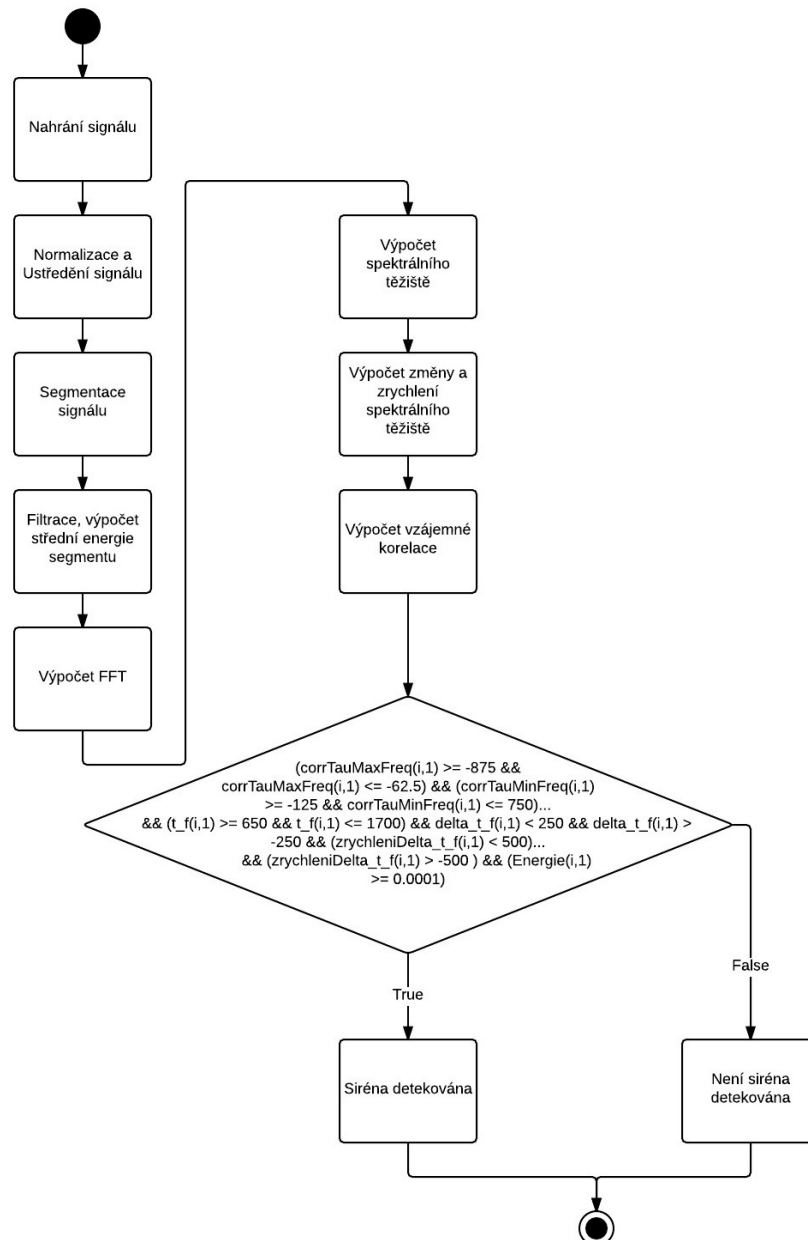
z vlastností signálu sirény. Pomocí podmínky *if* byla vytvořena rozhodovací část detekce signálu. Pokud je podmínka splněna, tak je segment signálu vyhodnocen jako *True* a je zapsán do proměnné *sirena*.

```
if(corrTauMaxFreq(i,1) >= -875 && corrTauMaxFreq(i,1) <= -62.5) &&
(corrTauMinFreq(i,1) >= -125 && corrTauMinFreq(i,1) <= 750) && (t_f(i,1) >= 650
&& t_f(i,1) <= 1700) && delta_t_f(i,1) < 250 && delta_t_f(i,1) > -250 &&
(zrychleniDelta_t_f(i,1) < 500) && (zrychleniDelta_t_f(i,1) > -500) && (Energie(i,1)
>= 0.0001)
```

Tab 2.: Tabulka mezních hodnot pro detekci sirény v záznamu zvuku

Parametry	Spodní hranice	Horní hranice
corrTauMaxFreq (Hz)	-875	-62.5
corrTauMinFreq (Hz)	-125	750
t_f (Hz)	650	1700
delta_t_f (Hz)	-250	250
zrychleniDelta_t_f (Hz)	-500	500
Energie	>= 0.0001	

Výsledný profil algoritmu pro detekci sirény demonstruje obrázek č. 22 a kopíruje kroky při zpracování signálu. Jde o popis offline detekce signálu, při níž se nahraný záznam krok pro krok zpracovává a nakonec vyhodnotí, zda v jeho částech byla detekována siréna.



Obrázek 22.: Schéma propojení bloků v Simulinku

Navržený algoritmus pro detekci sirény v prostředí MATLAB začíná nahráním signálu. Nahráný signál je normalizován a ustředěn. Pro rychle zpracování a vyhodnocení je signál segmentován pomocí Hammingovy okenní funkce. Šířka okna $w(n)$ je $l=128$ vzorků, tedy signál je rozdělen na 16 ms segmenty. Na všechny segmenty je aplikován filtr typu pásmová propust'. Spolu s filtrací byly vypočteny energie jednotlivých segmentů. Pro převedení segmentů z časové do frekvenční oblasti je použita rychlá Fourierova transformace. U jednotlivých spekter je vypočteno spektrální

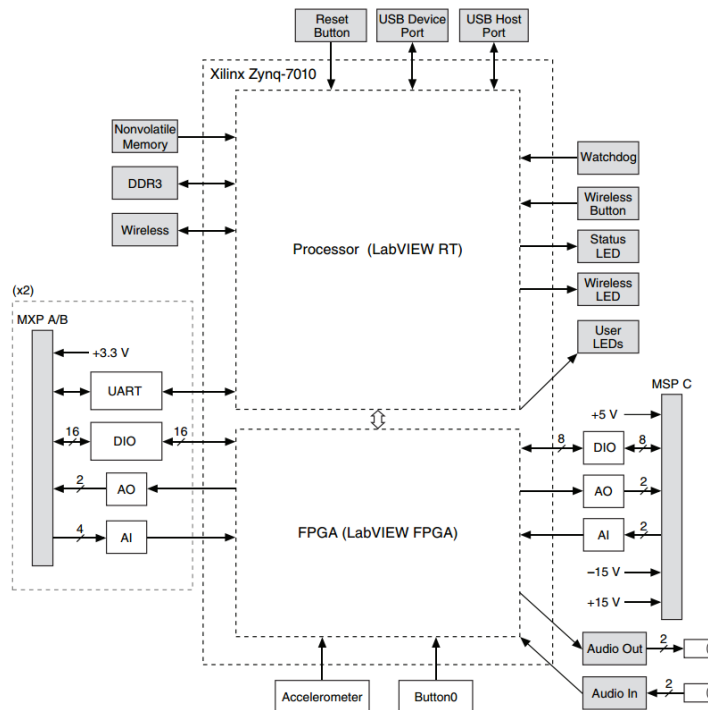
těžiště pro zjištění, na kterých frekvencích je soustředěno největší množství energie. Pro rozlišení signálů s rychlými změnami frekvence je vypočtena změna a průměrné zrychlení těžiště spektra. Lze říct, že jde o relativní rychlost růstu (klesání) frekvence těžiště spektra. Z prvotní analýzy signálu sirény byly vybrány spektra obsahující nejvyšší (nejnižší) frekvenci obsažené v siréně. Tato spektra jsou použita v posledním bloku, který vypočítává vzájemnou korelaci. Přesněji všechna spektra jsou korelována s nejvyšší a nejnižší frekvencí obsaženou v siréně. V poslední části diagramu je podmínka, která byla vyhodnocuje všechny vypočtené parametry nahraného signálu s hranicemi vytvořeny na základě analýzy záznamu sanitní houkačky.

6 Implementace detekčních metod v LabVIEW na platformě myRIO

Po prvotním zpracování záznamu sirény v prostředí MATLAB a vytvoření postupů algoritmu detekce sirény bylo třeba převést algoritmus do online zpracování. V tomto stavu vývoje bylo ideální vytvořit algoritmus detekce na platformě, která podporuje rychlý vývoj „rapid development“. Pro tento úkol jsme využili platformy myRIO od společnosti National Instrument a programovacího prostředí LabVIEW. LabVIEW je postaveno na grafickém programovacím jazyku, který umožňuje rychlou a intuitivní orientaci v kódu. Jednotlivé funkce jsou reprezentovány v bločcích, které se propojují mezi sebou pomocí „wire“ vodičem.

6.1 Platforma NI myRIO

Platforma myRIO je vestavné hardwarové zařízení založené na technologii RIO (reconfigurable I/O). NI myRIO obsahuje nejnovější technologii v podobě system on a chip (SoC) čipu Zynq® od společnosti Xilinx. Tento čip v sobě spojuje dvoujádrový procesor ARM Cortex-A9 a obvod FPGA. Systém NI myRIO také obsahuje 10 analogových vstupů, šest analogových výstupů, zvukové vstupně výstupní kanály a 40 digitálních vstupně výstupních linek. Dále zahrnuje WiFi, tříosý akcelerometr a několik programovatelných indikačních LED. NI myRIO programování v mnoha vývojových prostředích, mezi která patří LabVIEW a C/C++.[11]



Obrázek 23.: Schéma propojení bloků v Simulinku [11]

6.2 Přepočítání audio raw hodnoty na napětí

Pro přepočítání surových hodnot na napětí je třeba provést jednoduchý převod. Audio vstup používá 12bitový A/D převodník. Převod je přebrán z datasheetu zařízení myRIO.

$$V = \text{Raw data} * \text{LSB Weight}$$

$$\text{LSB Weight} = \text{Nominal Range} \div 2^{\text{ADC Resolution}}$$

Raw data jsou hodnoty získané z obvodu FPGA. *LSB Weight* je napěťový krok z rozsahu převodníku a napětí. *Nominal Range* je absolutní napěťová hodnota peak-peak vstupního kanálu.

$$\text{LSB Weight} = 5V \div 2^{12} = 1,221 \text{ mV}$$

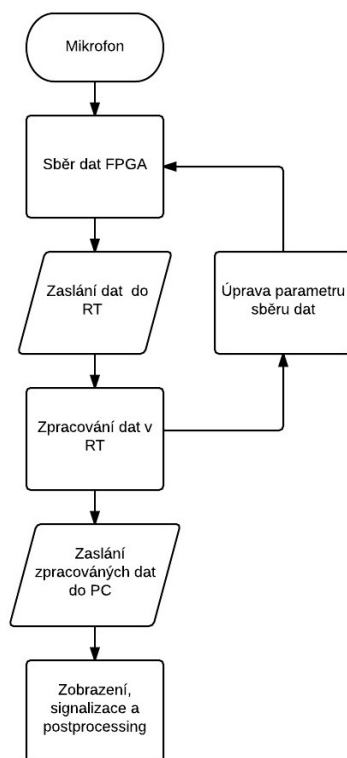
$$\text{Maximum Positive Reading} = +2047 * 1,221 = 2.499 \text{ V}$$

$$\text{Maximum Negative Reading} = -2048 * 1,221 = -2.5 \text{ V}$$

Maximum Negative Reading, Minimum Negative Reading jsou maximální hodnoty napětí na audio vstupu.

6.3 Měřicí řetězec pro zpracování záznamu zvuku v reálném čase

Na obrázku č. 24 můžeme vidět blokový řetězec měření na platformě myRIO. Tento řetězec propojuje tři nezávislé části, které si předávají parametry pomocí datových streamů, nebo po přímém přístupu do paměti (DMA). Tento řetězec zahrnuje předávání dat z FPGA modulu do Real-Time (RT) modulu a naposled do PC. Hlavní surová data jsou sbírána na FPGA a v určitý okamžik posílána do RT modulu. Vzorkovací frekvence signálu je $f_{vz} = 8000$ Hz. Podle Shannon-Kotelnikova teorému, kde je podmínka dodržena $f_{vz} > 2 \cdot f_{max}$. Jsou surová data posílána do RT modulu každých 16 ms hodnota odpovídá 128 vzorkům. RT modul na těchto datech provádí signálové zpracování a vyhodnocení naměřených parametrů. RT modul umožňuje nastavovat parametry na FPGA. V poslední řadě jsou zpracovaná data posílána pomocí datových streamů do hostujícího PC, které parametry vykresluje a případně vykonává postprocessing.



Obrázek 24.: Měřicí řetězec pro online zpracování signálu

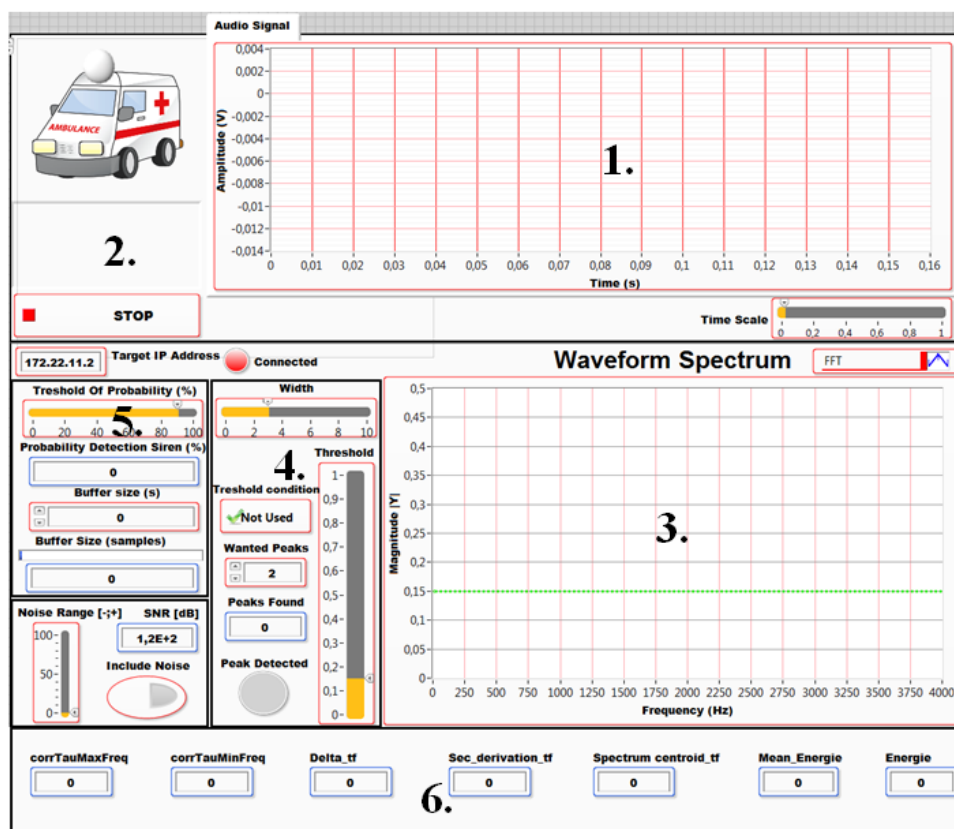
6.4 Grafická uživatelská aplikace

Celá aplikace je vytvořena jako LabVIEW projekt s názvem *myRecorder.lvproj*. V tomto projektu jsou umístěny všechny potřebné složky se subVI (blokovými funkcemi).

Pro spuštění hlavní PC aplikace je třeba otevřít VI s názvem *PC_app.vi*. V tomto VI je vytvořen hlavní panel aplikace pro zobrazení zpracovaného zvuku. Je rozdělen do několika sekcí, v nichž se nachází příslušná funkcionality.

Hlavní panel *PC_app.vi*

Hlavní panel aplikace *PC_app.vi* obsahuje různé funkční sekce. Okna s číselnými parametry jsou rozdělena do další podskupiny na výstupní (modře) a vstupní (červeně) hodnoty. Na obrázku č. 25 je zobrazen uživatelský panel.



Obrázek 25.: Schéma propojení bloků v Simulinku

6.4.1 Zobrazení zvukového záznamu v čase

V části 1 se jedná o zobrazovací část panelu, která vykresluje zvukový záznam v čase. Časovou osu x lze nastavovat pomocí posuvníku s názvem *Time Scale*. Osa y je prezentována v napětí. Toto napětí je vypočteno z hrubých hodnot získaných z FPGA.

6.4.2 Informační panel detekce sirény

Část 2 obsahuje informaci o detekci sirény v reálném čase (obr. 26). Signalizuje opticky a zvukově detekovanou sirénu. Zvukový záznam obsahující sirénu je přehrán pomocí reproduktoru PC.



Obrázek 26.: Informační panel detekce sirény

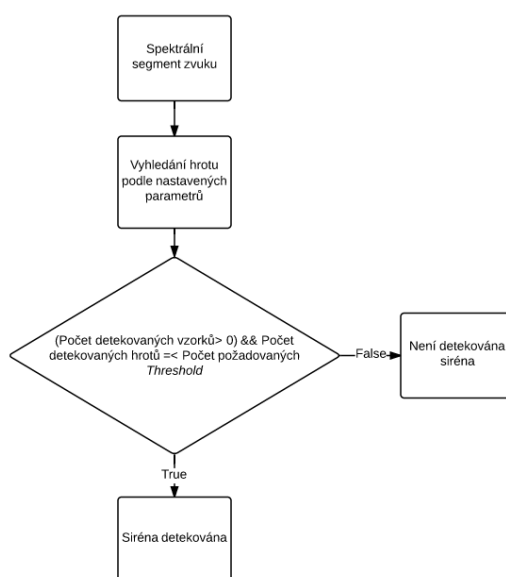
6.4.3 Amplitudové spektrum signálu

Blok číslo 3 zobrazuje amplitudové spektrum v reálném čase. Graf je doplněn o korelační spektrum s nejvyšší (nejnižší) obsaženou frekvencí sirény. Lze pozorovat pohyb spektrálních složek spektra po frekvenční ose x .

6.4.4 Hrotová detekce amplitudového spektra

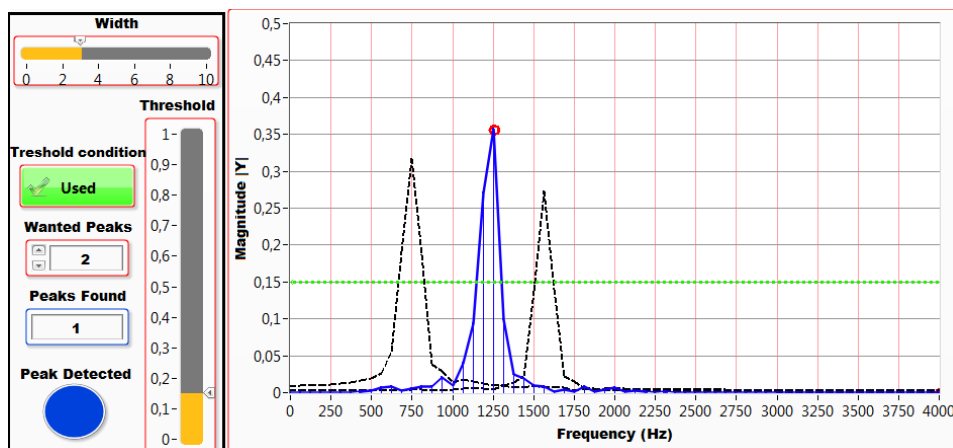
Panel 4 umožňuje použít přídatnou rozhodovací úroveň algoritmu detekce sirény. Pomocí tlačítka *Threshold condition* zapínáme/vypínáme rozhodovací podmínku. V okně *Wanted Peaks* se zadává maximální množství vzorků, které mají přesáhnout nastavenou hladinou *Threshold*. Pomocí posuvníků *Width a Threshold* nastavujeme počet vzorků a výšku hladiny detekce. Jednoduchým blokovým diagramem je vysvětlen postup detekce (obr. 27).

Vstupní parametr pro servisní funkci „peak detection“ je spektrální segment zvuku. Podle nastavených parametrů *Width a Threshold* vyhledá odpovídající počet hodnot, které splňují tyto parametry, ale následnou podmínkou se ptá, zda počet vyhledaných hodnot nepřesahuje počet požadovaných hodnot *Threshold*. Pokud jsou podmínky splněny, tak je výsledek hodnocen jako True.



Obrázek 27.: Schéma hrotové detekce

Na obrázku č. 28 je zachycena průběhu zpracování zvukového záznamu sirény. Modrá spektrální křivka odpovídá segmentu zpracovanému v reálném čase. Na vrcholu spektrálních čar je zachycena indikace hrotu, který odpovídá nastaveným parametrům. *Threshold* hladina je nastavena na hodnotu 0,15 a požadovaný počet detekovaných hrotů je nastaven na hodnotu 2.

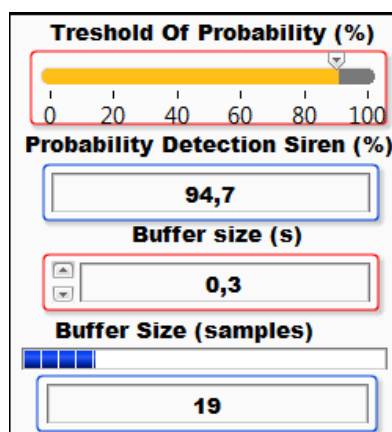


Obrázek 28.: Ukázka zpracování signálu v reálném čase

Černé čárkované křivky jsou korelační spektra, s nejvyšší (nejnižší) obsaženou frekvencí sirény. Jsou zde zobrazeny pro vizualizaci hranic detekce sirény.

6.4.5 Průměrný počet detekcí sirény v časovém úseku

Tato sekce umožňuje nastavovat procentuální hladinu úspěšných detekcí z celkového požadovaného počtu vyhodnocených segmentů signálu. Pomocí posuvníku *Threshold of probability* (obr. 29) můžeme nastavit požadovanou procentuální hladinu úspěšných detekcí sirény za definovaný čas.



Obrázek 29.: Panel s přednastavenými parametry pro vyhodnocení průměrného počtu detekcí

Probability Detection Siren (obr. 29) je hodnota, která představuje kolik procent z počtu vyhodnocených segmentů, bylo detekováno jako siréna. Vstupní hodnota *Buffer size(s)* je parametr v sekundách. Určuje, jak dlouhý má být časový úsek pro vyhodnocení detekce sirény. *Buffer size (samples)* je pouze přepočet časového úseku na počet segmentu, z něhož se vypočítá průměrná hodnota. Pro přepočet segmentů je uveden v následujícím vztahu (vzorec č. 8):

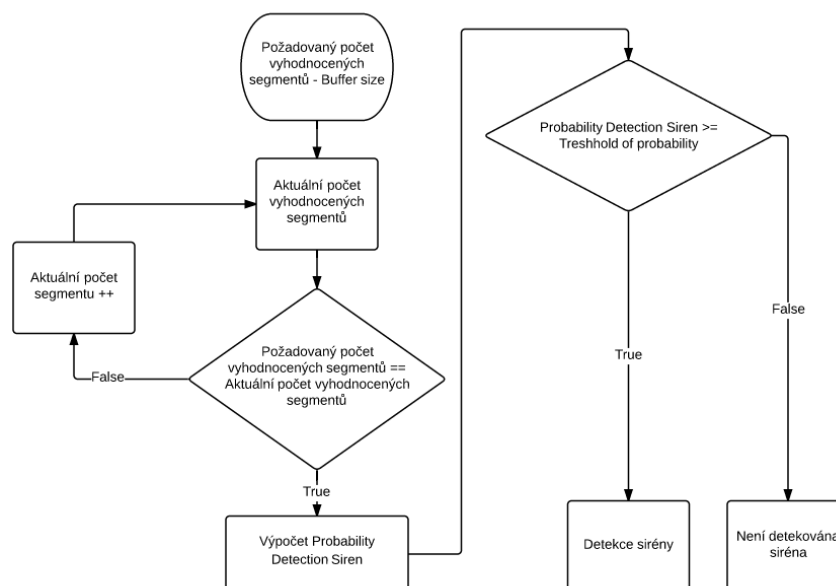
$$Buffer\ size(samples) = \frac{Buffer\ size(s)}{\frac{Délka\ segmentu\ (l)}{Vzorkovací\ frekvence\ (Hz)}} \quad (8)$$

$$Buffer\ size(samples) = \frac{0,3}{\frac{128}{8000}} \approx 19\ segmentů$$

Ve vzorci číslo 9 je uveden průměrný počet úspěšných detekcí sirény. Pomocí této podmínky jsme schopni v reálném čase měřit, kolik procent segmentů je falešně detekováno. Podmínka nám definuje procentuální hladinu počtu úspěšných detekcí proto, abychom mohli říct, zda se opravdu jedná o sirénu v daném časovém úseku.

$$Probability\ Detection\ Siren = \frac{\sum Segmenty\ detekovány\ jako\ siréna}{Celkový\ počet\ vyhodnocených\ segmentů} * 100 = (\%) \quad (9)$$

Na obrázku č. 30 je zobrazeno blokové schéma postupu při výpočtu průměrného počtu detekcí sirény za určitý časový úsek.



Obrázek 30.: blokové schéma postupu při výpočtu průměrného počtu detekcí sirény za určitý časový úsek

6.4.6 Výpis jednotlivých vypočtených parametrů

Poslední šestá část uživatelské aplikace zobrazuje vypočtené parametry segmentů signálu. Jde o parametry, které jsou vyhodnocovány v algoritmu pro detekci signálu.

6.5 Navržený detekční algoritmus pro reálné zpracování signálu

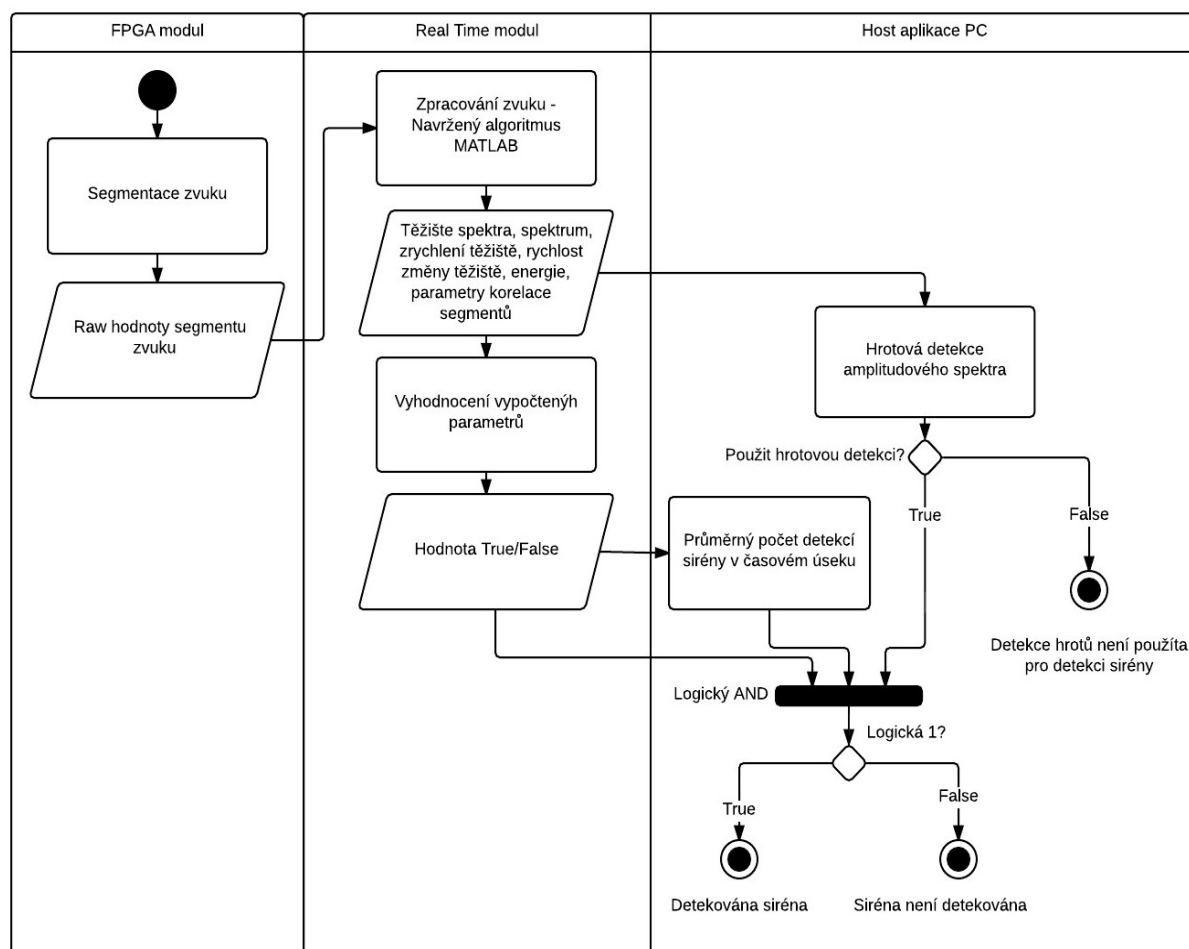
Výsledný algoritmus pro detekci zvuku vychází z prvotního návrhu v prostředí MATLAB. Jednotlivé kroky při zpracování zvuku odpovídají blokovému schématu pro MATLAB.

Tab 3.: Tabulka mezních hodnot pro detekci sirény v reálném čase

Parametry	Spodní hranice	Horní hranice
corrTauMaxFreq (Hz)	0	812,5
corrTauMinFreq (Hz)	-812,5	62,5
t_f (Hz)	800	1600
delta_t_f (Hz)	-200	200

zrychleniDelta_t_f (Hz)	-200	200
Energie	≥ 0.15	

V tabulce č. 3 lze vidět jednotlivé parametrové hranice pro detekci sirény. Vycházeli jsme z vypočtených parametrů při analýze záznamu sirény. Po prvotním testování aplikace byly parametry změněny. Parametry je možné individuálně nastavit ve funkčním bloku VI s názvem *Decision_Algorithm(SubVI).vi*.



Obrázek 31.: Blokový diagram komunikace jednotlivých modulů pro zpracování a detekci signálu v reálném čase

Na obrázku č. 31 vidíme blokový diagram komunikace jednotlivých modulů pro zpracování a detekci signálu v reálném čase. Zaznamenáním zvuku na audio vstupu a jeho navzorkováním jsou jednotlivé segmenty surových hodnot přeneseny pomocí

DMA bufferu do RT modulu. V tomto modulu jsou segmenty signálově zpracovány podle navrženého algoritmu z prostředí MATLAB. Vypočtené a vyhodnocené parametry segmentu jsou zaslány do uživatelské aplikace na PC. V PC aplikaci je na vypočtených parametrech vykonán post-processing. Pokud se všechny vyhodnocené parametry pohybují v požadovaných mezích, tak je záznam zvuku vyhodnocen jako siréna a uživatel je na tuto situaci upozorněn.

7 Experiment

V této části bylo nutné otestovat algoritmus pro detekci sirény v reálných podmínkách.

Použité vybavení:

2× notebook

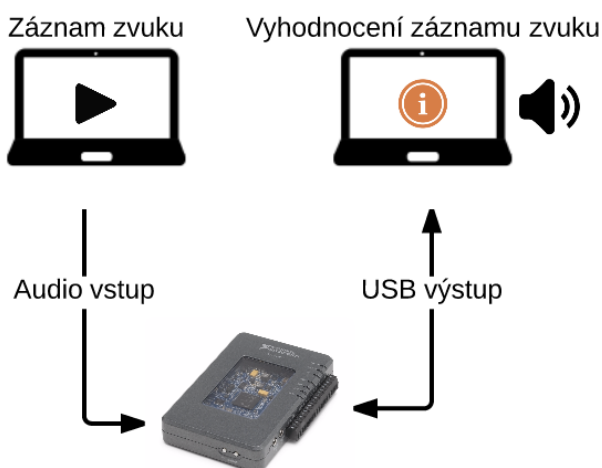
myRIO

audio kabel, USB kabel

zvukové záznamy

7.1 Zapojení testovacího řetězce

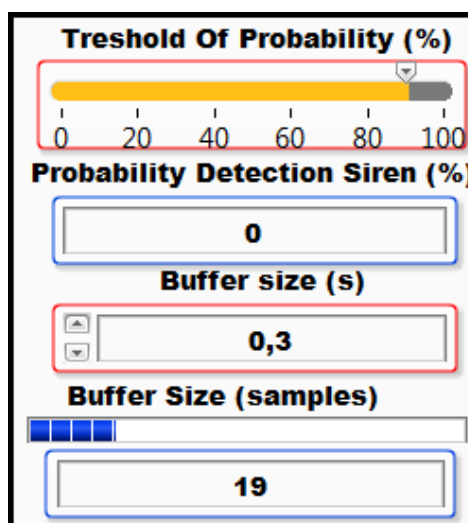
Měřicí řetězec (obr. 32) se skládá z platformy myRIO, dvou přenosných počítačů a propojovacích kabelů. Platforma myRIO je připojená pomocí USB kabelu k počítači, který obsahuje uživatelskou aplikaci *PC_app.vi* pro zobrazení a vyhodnocení analyzovaných dat. Zvukový výstup druhého počítače je pomocí audio kabelu připojen na audio vstup zařízení myRIO. Počítač propojený audio kabelem se zařízením myRIO je objektem přehrávající zvukové záznamy.



Obrázek 32.: Blokové schéma testovacího řetězce

7.2 Průběh měření

Po zapojení měřicího řetězce bylo na počítači pro přehrávání zvuku vybráno několik různých typů záznamu zvuku. Jde o složení zvuků obsahujících hluk dopravy, troubení, lidské řeči, motorů dopravních prostředků atd. Poté byla počítači pro vyhodnocení a detekci sirény spuštěna aplikace *PC_app.vi*. Spuštěním aplikace jsme ve smyčce spustily záznamy zvuků a vyhodnocovaly falešné nebo pozitivní detekce sirény.



Obrázek 33.: Testovací parametry

Parametry uživatelské aplikace jsou nastaveny na hodnoty $Buffer\ size = 0,3\ s$ to znamená, že průměr úspěšných detekcí signálu bude počítán z času 300 ms. Plovoucím parametrem je *Threshold Of Probability*. Tato hladina bude určovat přísnost výsledné detekce časového úseku o velikosti 300 ms. Například pokud bude nastavena hladina *Threshold Of Probability* na hodnotu 100 % znamená to, že všech 19 segmentů musí být detekováno jako siréna. Pokud tomu tak není, tak není signalizována detekce sirény.

7.3 Výsledky experimentu

Algoritmus byl testován na 11 reálných záznamech zvuku. Z toho 4 obsahovaly záznam sirény. Hodnoty k jednotlivým záznamům říkají, kolik procent z celkové doby záznamu, bylo detekováno jako siréna. V kolonce procentuální úspěšnost je hodnota průměrné úspěšnosti detekce signálu. Zeleně vyznačené hodnoty ukazují na úspěšné vyhodnocení záznamu signálu. Detekce signálu v běžném provozu vykazuje hodnotu okolo 90% úspěšné detekce. Samozřejmě se snižující se hladinou průměrného počtu detekcí v daném časovém intervalu se zvyšuje šance falešné detekce. Tento trend lze vidět u hodnot, které jsou v tabulce vyznačeně červeně.

Tabulka popisuje výsledky experimentu na různých hladinách *Threshold Of Probability*.

Tab 4.: Tabulka mezních hodnot pro detekci sirény v záznamu zvuku

Názvy záznamů zvuků	Hladiny požadované přesnosti detekce signálu v definovaném časovém intervalu						Procentuální úspěšnost detekce (%)
	100%	90%	80%	70%	50%	30%	
Nahrávka sirény WAIL	99	100	100	100	100	100	99,8
Sanitka_houknuti	90	95	100	100	100	100	97,5
Sanitka_krizovatka	90	95	95	98	98	99	95,8
Sanitka_prujezd	85	90	95	98	100	100	94,7
Dopruch1	0	0	0	0	0	0	0,0
Motor	0	0	0	0	0	0	0,0
Hudba	0	0	0	0	0	0	0,0
Prujezd_bus	0	0	0	0	0	0	0,0
Venku_ptak	0	0	0	0	0	0	0,0
Vrtulnik	0	0	0	0	0	0	0,0
Troubeni	0	10	13	17	20	25	14,2
Hláška A	0	0	50	100	100	100	58,3
Festival v ulicích	0	0	0	10	20	50	13,3

8 Závěr

Ve své diplomové práci jsem se zabýval vytvořením algoritmu pro detekci houkačky sanitního vozu v silničním provozu. Průzkumem stavu současných metod detekce jsem zjistil, jakým směrem se vývoj této problematiky zaobírá. Protože jde se jedná aplikaci v automobilovém průmyslu, je třeba si uvědomit, v jakém prostředí bude siréna detekován, a také to, jakým způsobem je možné minimalizovat výpočetní náročnosti celého algoritmu. Pro správný návrh algoritmu jsem se seznámil se sirénou sanitního vozu.

Volba použití matematických metod byla posuzována podle jejich náročnosti na výpočetní výkon. Celý postup při analýze záznamu sirény byl vypracován v souladu s požadavky na výpočetní výkon. V první části diplomové práce byl zvukový záznam vyhodnocen v časové a spektrální oblasti pomocí programu MATLAB. Díky jeho charakteristickým vlastnostem byly vybrány metody pro zpracování signálu, které nejvhodněji vyhodnocují charakteristické vlastnosti signálu. Mezi tyto metody patřily segmentace, FFT analýza, výpočet spektrálního těžiště, střední krátkodobé energie, změna frekvence spektrálního těžiště, relativní zrychlení spektrálního těžiště a křížová korelace signálu. Z vypočtených veličin byly vybrány mezní parametry, které jsou charakteristické pro zvuk houkačky sanitního vozu. První návrh algoritmu byl otestován v prostředí MATLAB pomocí offline zpracování záznamu zvuku.

Implementace algoritmu pro vyhodnocení reálných podmínek proběhla na platformě myRIO. Pomocí programového prostředí LabVIEW jsem platformu naprogramoval a vytvořil uživatelský panel, který zobrazuje v reálném čase spektrální okno segmentu signálu a vypočtené parametry zaznamenaného zvukového signálu. Výsledný algoritmus a mezní parametry pro zpracování signálu v reálném signálu jsem převzal z návrhu v prostředí MATLAB. Během vývoje aplikace jsem algoritmus a mezní parametry pozměnil podle vypořizovaných parametrů při zpracování čistého záznamu signálu. V závěru práce jsem otestoval detekční schopnost algoritmu na reálných záznamech zvuku. Výsledky testů poukazují na velkou detekční schopnost algoritmu v reálných podmínkách.

Návrh pro budoucí vývoj detekčního zařízení by mohl spočívat v projektu vstupně analogového zařízení pro zpracování zvuku, návrhu aplikace algoritmu na

embedded platformě a dalšího testování v reálných podmínkách s hledáním ideálních mezních parametrů.

9 Literatura

- [1] *Akustická pěna* [online]. FRYČ, Alexandr. [cit. 2015-05-21]. < <http://www.akusticka-pena.cz/files/136.jpg> >
- [2] TŮMA, J. *Zpracování signálů získaných z mechanických systémů užitím FFT*. Praha: Sdělovací technika, 1997, vii, 174 s. ISBN 80-901936-1-7.
- [3] JURÁK, Petr. *SIGNÁLOVÁ ANALÝZA MLUVENÝCH SOUHLÁSEK*. < https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=86690 >. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Prof. Ing. MILAN SIGMUND, CSc.
- [4] MIELKE, M. a kol. *Integrated Circuit for Detection of Acoustic Emergency Signals in Road Traffic*. Institute of Microsystems Technology University of Siegen, Siegen, Germany, 2010, : 4.
- [5] ROMANOWSKI, A. a kol. Method for Automatic Driver Notification about Detection of the Emergency Vehicles. *Automatyka*
- [6] FAZENDA, B. a kol. *Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems*. School of Computing and Engineering, University of Huddersfield, Huddersfield, UK, 2009, : 6.
- [7] *Humusoft: Jazyk pro technické výpočty* [online]. [cit. 2015-04-01]. < <http://www.humusoft.cz/produkty/matlab/matlab/> >
- [8] *Ambulance Visibility: The effects of soundproofing in modern vehicles* [online]. [cit. 2015-04-01]. < http://ambulancevisibility.com/index.php?p=1_12 >
- [9] ČERNOCKÝ, H. *Zpracování řečových signálů — studijní opora* [online]. 2006 [cit. 2015-07-21]. Vysoké učení technické v Brně.
- [10] PSUTKA, J. *Komunikace s počítačem mluvenou řečí*. Vyd. 1. Praha: Academia, 1995, 287 s. ISBN 80-200-0203-0.
- [11] User Guide and Specifications: NI myRIO-1900 [online]. [cit. 2015-03-20]. < <http://www.ni.com/pdf/manuals/376047a.pdf> >
- [12] *Prime Safety, s.r.o.: Sirény a reproduktory* [online]. [cit. 2015-04-01]. < <http://www.prime-safety.com/sireny-a-reproduktory.html> >

10 Přílohy

Příloha I.	Zdrojový kód MATLAB
Příloha II.	Odkazy na zvukové záznamy
Příloha CD	Projekt v LabVIEW, zvukové záznamy

Příloha I.

Zdrojový kód MATLAB

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Sound Analysis with MATLAB Implementation   %
%                                           %
% Author: Josef Paleček 04/01/15           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear all;
close all;
clc;
clear t_f;
```

```
%% Nahrani signalu
load spektrummin2;
load spektrummax2;
```

```
[signal,fs] = audioread('C:\.\us_wail60s.mp3');
```

```
[pocetvz,sloupcu]= size(signal);
plot(signal);
```

```
if sloupcu > 1
signal(:,2) = [];
end
```

```
if fs ~= 8000
%% Resample
signal=resample(signal,8000,fs);
fprintf('vzorkovaci frekvence 1. %d\n',fs);
fs=8000;
end
fprintf('vzorkovaci frekvence %d\n',fs);
[pocetvz,sloupcu]= size(signal);
```

```
%% Normalizace a ustredeni signalu
```

```

signal=signal-mean(signal); %ustredeni
signal=signal/max(abs(signal)); %normalizace
casovazakl = (0:1/fs:(length(signal)-1)/fs);

fprintf('delka zaznamu %4.2f s\n',(max(casovazakl)));

%% Pridani sumu do signalu - pomoci pomeru snr
% signalBezSumu = signal;
% signal = awgn(signal,0,'measured'); % Add white Gaussian noise.
% plot(casovazakl,signalBezSumu,casovazakl,signal) % Plot both signals.
% legend('Original signal','Signal s Gausovym sumem');
%
% figure();
% plot(casovazakl,(signal - signalBezSumu));
% title('Bílí šum');

%% Filtrace
Fp = [500 2000]; % propustne pasmo
Fs = [350 2300]; % frekvence utlumeni
Fvz = fs; % vzorkovací frekvence
Ap = 0.1; % zvlneni v propustnem pasmu [dB]
As = 60; % potlacení mezi Fp a Fs [dB]
d=fdesign.bandpass(Fs(1),Fp(1),Fp(2),Fs(2),As,Ap,As,Fvz);
%navrh filtru o danyh vlastnostech
hd=design(d,'butter');

%vykreslení signalu
figure();
title('\bf Nahravka sireny WAIL');
plot(casovazakl,signal);
xlabel('Cas[s]');
ylabel('Normovana Amplituda');
%% Vytvoreni Okenni funkce
%pri vz. frekvenci 8000 == 2^10 pro funkci fft musi byt 2^N bude delka
%segmentu 128 (16ms) vzorku, 8000/128 * 31,25Hz frek skok

delkaseg=round(fs*16*10^(-3));
pocetseg=floor(pocetvz/delkaseg);
f_bin=0:delkaseg-1;
f = f_bin * fs/delkaseg;
okno_hamm=hamming(delkaseg);

```

```

%% Segmentace signalu hammingovym oknem
for i=1:pocetseg
    DATA(i,:)=signal(((i-1)*delkaseg+1):i*delkaseg,1).*okno_hamm; %segmentace
    % DATA_2(i,:)=DATA(i,:);
    DATA(i,:)=filter(hd,DATA(i,:));
    Energie(i,:)=(1/delkaseg)*sum(DATA(i,:).*DATA(i,:)); %stredni kratkodoba energie
end
StredniHodnota = mean(Energie); %prumer energie signalu

%% Vypocet spektra a teziste spektra
delta_t_f(1,1)=0;
zrychleniDelta_t_f(1,1)=0;

for i=1:pocetseg
    spektrum(i,:)=(2*((1/delkaseg)*(abs(fft(DATA(i,:)))))); %vypocet spektra
    %spektrum(i,:)=(2*((1/(pocetvz/2))*(abs(fft(DATA(i,:))))));
    teziste(i,1)=(sum((spektrum(i,1:delkaseg/2)).*(0:((delkaseg/2)-1))))/(sum(spektrum(i,1:delkaseg/2))); %vazeny prumer spektralni teziste
    t_f(i,1)=teziste(i,1).*(fs/delkaseg);

    if i >= 2 && i <= pocetseg
        delta_t_f(i,1)=(t_f(i,1) - t_f(i-1,1)); %rychlost zmeny frekvence teziste spektra
    if i >= 3 && i <= pocetseg
        zrychleniDelta_t_f(i,1) = (t_f(i,1) - 2*(t_f(i-1,1)) + t_f(i-2,1)); %druha derivace
        frekvence teziste spektra
    end
end
end

figure();
plot(delta_t_f);
title('\bfDerivace tezite spektra')
ylabel('f [Hz]');
xlabel('Pocet segmentu');

figure();
plot(zrychleniDelta_t_f);
title('\bfDruha derivace tezite spektra')
ylabel('f [Hz]');
xlabel('Pocet segmentu');

```

```
figure();
plot(f, spektrum((1:pocetseg),:));
xlim([0 4000]);
title("\bfJednostranné spektrum signálu")
xlabel('f [Hz]');
ylabel('|Y(t)|');
```

```
figure();
plot(f, spektrum(2567,:));
title("\bfSpektrum okna s minimalni obsazenou hodnotou frekvence")
xlim([0 4000]);
xlabel('f [Hz]');
ylabel('|Y(t)|');
hold on;
figure();
```

```
plot(f, spektrum(1458,:));
title("\bfSpektrum okna s maximalni obsazenou hodnotou frekvence")
xlim([0 4000]);
xlabel('f [Hz]');
ylabel('|Y(t)|');
```

```
figure();
plot(f, spektrum(2567,:));
title("\bfSpektra oken s maximalni a minimalni obsazenou hodnotou frekvence sireny")
xlim([0 4000]);
xlabel('f [Hz]');
ylabel('|Y(t)|');
hold on;
plot(f, spektrum(1458,:), 'r');
```

```
figure();
plot(t_f);
title("\bfTezite spektra")
ylabel('f [Hz]');
xlabel('Pocet segmentu');
%% Korelace jednostranneho spektra nahraného signálu a minimalnich, hodnot spektra
signálu
for i=1:pocetseg
```

```

[amplitudyMin(:,i),tauMin(:,i)] =
xcorr(spektrum(i,1:delkaseg/2),spektrummin2(1,1:delkaseg/2));
[amplitudyMax(:,i),tauMax(:,i)] =
xcorr(spektrum(i,1:delkaseg/2),spektrummax2(1,1:delkaseg/2));
end
[radky,sloupce]=size(tauMin);
%% Najdi největší(nejmenší) amplitudu ckrelace(min,max) a najdi k temto hodnotam
hodnotu frekvence posunu
%tau a relikii jsme ze korelace se nachazi kde jsme nasli maximalni
%amplitudu a vypsali jsme tau korelacnimu maximu
for i=1:pocetseg
[maxAmpcorr,vzorekAmpMin]=max(amplitudyMin(:,i));
corrTauMin(i,1)=tauMin(vzorekAmpMin,i); %hodnota tau vzorku
[maxAmpcorr,vzorekAmpMax]=max(amplitudyMax(:,i));
corrTauMax(i,1)=tauMax(vzorekAmpMax,i);
end
corrTauMaxFreq=corrTauMax.*(fs/delkaseg);
corrTauMinFreq=corrTauMin.*(fs/delkaseg);
%% Průměrný počet detekcí sirény v časovém úseku
velBuff=118;
casBuff=velBuff*16*10^(-3);
buffer=velBuff;
start = 1;
k=1;

fprintf('Velikost bufferu: %2.1fs \n',casBuff);
for i=1:pocetseg
    if (corrTauMaxFreq(i,1) >= -875 && corrTauMaxFreq(i,1) <= 125) &&
(corrTauMinFreq(i,1) >= -125 && corrTauMinFreq(i,1) <= 750)...
        %%if (corrTauMaxFreq(i,1) >= -875 && corrTauMaxFreq(i,1) <= -62.5) &&
(corrTauMinFreq(i,1) >= -125 && corrTauMinFreq(i,1) <= 750)...
            && (t_f(i,1) >= 650 && t_f(i,1) <= 1700) && delta_t_f(i,1) < 250 && delta_t_f(i,1)
> -250 && (zrychleniDelta_t_f(i,1) < 500)...
            && (zrychleniDelta_t_f(i,1) > -500 ) && (Energie(i,1) >= 0.0001)%%0.001
            sirena(i,1)=1;
        else
            sirena(i,1)=0;
        end
end

if (i == buffer)

```

```

sumaBuffer(k,1)=(sum(sirena(start:buffer,1))/velBuff);

    if ((sum(sirena(start:buffer,1))/velBuff) >= 0.90)

        fprintf('%d. Detekce sireny %2.1fs bufferu v case: [%4.2f, %4.2f]s\n'....
            ,k,casBuff,((buffer/velBuff)*(casBuff))-casBuff,(buffer/velBuff)*(casBuff))
        %16ms*60
        fprintf('%dFrekvence detekovana v cas %d\n',buffer);
    end
    k=k+1;
    start = start+velBuff;
    buffer = velBuff*k;

    if (buffer > pocetseg)
        buffer=pocetseg;
    end
end
end
end
%% Pridavne syntaxe pro nalezeni segmentu
%
% figure();
% plot(f, spektrum(326,:)); % segment pro min frekvenci
% hold on;
% plot(f, spektrum(820,:), 'r'); % segment pro maximalni frekvenci
% title('\bfJednostranne spektrum oken s max,min hodnotou frekvence segmentech')
% xlabel('f [Hz]');
% ylabel('Y(t)');
% xlim([0 4000])
% figure();
% plot(tauMin(:,326), amplitudyMin(:,326));
% hold on;
% plot(tauMax(:,820), amplitudyMax(:,820), 'r');
% title('\bfGrafy korelace signalu vzorku spektra s nejvyssi a nejnizsi frekvenci')
%
% for i=1:pocetseg
%     [x(i,1),y(i,1)]=max(spektrum(i,1:delkaseg/2));
%     if (x(i,1)==0.269467622763983)
%         findmax=i;
%     end
% end

```

Příloha II. Odkazy na zvukové záznamy

Sanitka_krizovatka

Odkaz: <https://youtu.be/QrW1smwU0yc>

Sanitka_prujezd

Odkaz: <https://youtu.be/WFq0oxAAiyw>

Troubení

Odkaz: <https://youtu.be/xBlGg77LMBo>